문제해결을 위한
창의적 알고리즘

KOI 교육교재 (초급)

한국정보화진흥원

집필진

전현석(경기과학고등학교)
배준호(경남정보고등학교)
문지은(충북과학고등학교)
김지혜(흥덕고등학교)
송석리(선린인터넷고등학교)
안문옥(단원고등학교)
정종광(경기과학고등학교)
남상천(경기과학고등학교)
최웅선(수원하이텍고등학교)
김영희(경남과학고등학교)
정상수(세종아름중학교)
정웅열(경기북과학고등학교)
경기도중등정보과교육연구회

머리말

컴퓨터는 우리 생활 어디에나 있으며, 인간의 삶을 풍요롭게 만들어가고 있다. 컴퓨터의 본질은 세상의 모든 것을 디지털로 저장하고, 그렇게 저장된 것을 다루어 인간이 필요로 하는 모든 것과 상상하는 모든 것을 실현해내는 큰 힘이라고 할 수 있다.

컴퓨터를 활용한 다양한 적용과 구현이 있지만, 역사적 발전과 흐름 속에서 컴퓨터의 막강한 계산 능력을 활용하는 문제(상황) 해결은 현재 인간의 삶을 획기적으로 바꾸어 놓은 본질로서의 의미를 가지고 있다.

컴퓨팅의 발전은 컴퓨터 하드웨어와 네트워크의 발전과 함께 인간 삶의 방식을 바꾸어 놓았고, 그러한 발전의 근본적인 기초/개념/원리는 보다 적은 계산으로 빠르고 정확한 계산 결과를 얻어내기 위한 인간의 창의적인 아이디어와 노력이라고 할 수 있다.

본 교재에서는 다양한 문제 상황에서 활용할 수 있는 효과적인 문제 해결 방법을 배울 수 있는 기초적인 개념들을 학습할 수 있는데 특히, 컴퓨터 없이 손가락과 생각만으로도, 매우 복잡한 상황들에서 원하는 답이나 해법을 효과적으로 얻어낼 수 있는 다양한 아이디어와 방법들을 학습할 수 있다.

직접적으로는 지난 수년간 다루어졌던 정보올림피아드 지필 기출 문제들을 다루지만, 그 안에는 다양한 문제 상황과 퍼즐, 복잡한 문제 상황들을 직간접적으로 다루는 방법과 아이디어들이 작성되어 있으며 프로그래밍언어를 사용한 컴퓨팅에 그러한 방법들을 사용해 효과적으로 다룰 수 있게 된다면, 이전까지는 생각하지 못했던 수많은 데이터를 아주 빠른 시간 이내에 분석하고 처리하여 우리가 상상하는 많은 것들을 이루어낼 수 있게 될 것이다.

본 교재에서 다루어지는 각 문제들에 대한 의의, 해설, 해법들은 그동안 학교 현장에서 직접 문제들을 다루면서 공교육 차원에서 학생들에게 교육해 온 여러 선생님들의 노력으로 만들어진 것으로, 원래 기출문제의 의도나 해법과 다를 수 있다.

또한, 제시되는 의미나 해법들은 매우 많은 가능한 해법들 중 하나의 예시일 뿐이기 때문에, 맹목적으로 암기하는 일은 없어야 할 것이다. 정보에서 다루어지는 문제 상황들과 그 해법들은 특별한 정답이라는 것이 존재하지 않는다고 할 수 있기 때문이다. 더 효과적인 아이디어와 방법을 활용하는 창의적인 문제 해결 방법들이 만들어질 수 있으며, 그러한 문제 해결 방법들과 아이디어들은 본 교재를 공부하는 학생들에게 남겨두고자 한다.

본 교재의 구성은 크게 3개의 부분으로 나뉘어져 있다.

첫 번째로 정보올림피아드 지필 문제의 평가 목표를 간단히 다루는데, 이는 정보올림피아드에서 다루어지는 문제들의 목적에 대해서 명확히 이해하기 위함이다. 정보올림피아드에서 다루어지는 문제들은 기본적으로 컴퓨터로 다룰 수 있는 계산 문제로서 그 범위나 크기가 매우 큰 것들로서, 일반적인 부르트포스 방법으로는 제약 조건(시간, 기억공간) 이내에 해결 할 수 없는 것들이 대부분이다. 따라서 그러한 제약 조건 이내에 효과적으로 해결할 수 있도록 하는 창의적인 문제해결 사고, 패턴/관계 분석, 재귀적 문제해결 방법 등을 이해하는 것을 목표로 하고 그러한 사고와 방법들이 구체적으로 구현되어있는 다양한 문제해결 코드를 이해하는 것에 중심을 맞추고 있다.

두 번째로 다양한 퍼즐/문제 상황에서의 문제해결 방법 설계 전략과 관련한 간단한 예시로서 2012년도 이전의 문제들을 분석하고 분류한 기초적인 문제들을 살펴본다. 이는 간단한 문제 상황으로 표현된 문제들을 대략적으로 분류하여 다루어 보는 것으로서 실제 실기 상황에서 다루어지는 복잡한 문제 해결 상황들에서 효과적으로 사용될 수 있는 기본적 능력을 키우는 것이라고 볼 수 있다.

세 번째로 2013년 이후의 실제 지필 문제들에 대한 예시 해법들을 다룬다. 2013년을 기준으로 지필 문제의 유형이 실제 문제해결에서 사용할 수 있는 기초/개념/원리를 다루는 형태로 바뀌었기 때문에, 2013년 이후의 필기 문제를 다루고 이해하는 것은 실기 문제를 해결하는데 있어서도 매우 유용할 것으로 생각된다.

마지막으로, 본 교재가 만들어질 때까지 많은 노력을 하시고, 많은 관심과 지원을 아낌없이 해주신 모든 분들께 감사드리며, 또 앞으로 이 교재를 공부하며 자라나 우리나라를 이끌어갈 많은 학생들에게 격려와 함께 감사의 말을 남기고자 한다.

저자 일동

목 차

I 정보올림피아드 지필 문제의 평가 목표

1. 정보올림피아드 지필 문제의 접근 방법

정보올림피아드 지필 문제들은 다음과 같은 평가 목표들을 가지고 있다고 볼 수 있다.

- 창의적 문제해결 사고
- 패턴분석 및 관계분석
- 코드/함수의 이해 및 의미 분석
- 재귀 함수의 이해
- 문제해결 프로그래밍 코드의 이해

이러한 평가 목표들은 실제로 프로그래밍언어를 활용하는 문제해결을 위해 가장 기초적인 능력들을 평가하는 것으로서, 다음과 같은 일반적인 사고방식과 접근 방식으로는 모든 문제들을 주어진 제한 시간 이내에 절대로 해결할 수 없다.

- 무조건 다 해 본다.
- 변수들의 변화와 값의 변화를 모두 따라간다.
- 문제 풀이 방법은 하나뿐인 것으로 생각한다.
- 문제 풀이 방법을 외운다.

따라서, 주어지는 문제들을 효과적으로 해결하기 위해서는

- "문제의 의도가 무엇일까?"
- "모든 경우를 해보는 것은 절대로 아니다. 적어도 매우 효과적인 한 가지 방법은 있을 것이다."
- "복잡해 보이는 코드는 단순히 꼬아 놓은 것이 아니고, 어떤 의미를 전달하기 위해 주어진 샘플 코드이다.“

라고 생각하는 것이 맞다.

이런 문제들에 대해서 경험해 보지 않은 사람들은 반드시 초등부 문제부터 풀이해 보는 것이 좋다. 왜냐하면 중고등 문제들은 초등 문제들의 확장(더 복잡, 수의 크기나 범위가 더 커짐)이라고 볼 수 있기 때문이다.

2. 문제해결 프로그래밍 코드의 이해

문제해결 프로그래밍의 가장 중요한 목표는 주어진 문제 상황에서 컴퓨팅 능력을 활용해 필요로 하는 정확한 답이나 해법을 주어진 제한 시간(대부분 1초) 이내에 얻어낼 수 있도록 하는 것으로 문제해결 프로그래밍의 계산량이 가장 중요하다고 볼 수 있다.

지필 문제의 마지막 문제들은 이전 년도의 실기 문제 중에서 학생들이 가장 어려워 했거나 풀어내지 못했던 문제들에 대한 모범 코드와 해법을 제시하는 형태의 문제들이 나오는데, 가장 기본적인 형태는 재귀함수를 이용한 다중 가지치기(backtracking), 백트랙킹 시간 줄이기(branch & bounding) 등이나 배열 채워나가기인 동적프로그래밍(dynamic programming) 코드가 주어지는 경우가 많다.

재귀함수를 사용하면 가능한 모든 경우에 대해서 깊이우선 탐색방법으로 탐색할 수 있지만, 대부분은 실기 대회에서의 문제별 제한 시간 이내에 풀어낼 수 없다. 그렇기 때문에, 문제해결에 필요한 계산량(time complexity)이 적은 다차원 동적테이블을 활용한 문제해결 코드의 예시가 제시되기도 한다.

마지막 문제들은 주어진 문제 상황을 해결할 수 있는 일종의 예시 코드를 보여주고

- "이런 식으로도 할 수 있다"
- "이해해 보렴.."

등의 의미를 가진다고 볼 수 있다.

지필 평가의 마지막 부분 문제들은 실기 대회를 위한 전초전 형태로서, 따로 분리시켜서 실제 실기 형태로 직접 연습해보는 것도 좋은데, 실기 대회에서의 문제해결은 다음과 유사한 과정이 필요하다고 할 수 있다.

- 문제 상황에 대한 이해
- 문제 상황에서 주어진 정보들에 대한 추상화(변수, 함수, 값 등)
- 문제 상황에서 최종적으로 얻고자하는 목표(정확한 값과 제한 시간) 인식
- 주어진 제한 시간 이내에 정확한 목표를 얻어내기 위한 창의적인 사고
  (기본적으로는 부르트포스 사용, 시간 초과를 생각해서 계산량을 줄여야 함)
- 생각한 방법을 프로그래밍, 샘플 데이터를 넣어보고 확인
- 샘플 데이터가 잘 된다면? 안 되는 경우(반례)가 있을지 충분히 확인 후 제출

샘플 데이터를 입력해 자신이 만든 문제해결 프로그래밍 코드들을 검증해 볼 때에는 일반적으로 가장 작은 데이터, 가장 큰 데이터, 모두 같은 데이터, 입력 범위의 양끝 단에 있는 임계 데이터, 오름차순 데이터, 내림차순 데이터, 오버플로, 언더플로 등을 가능한 많이 생각해 보는 것이 좋다.

퍼즐 해결

대회에서 주어지는 문제들을 해결하기 위해서는 논리적인 사고와 다양한 생각, 창의적인 아이디어를 필요로 한다. 하지만 다양한 형태의 문제를 접하고 풀어보는 과정에서 그러한 능력들이 자연스럽게 길러질 수 있다.

주어지는 문제들을 해결하기 위해 특별한 수학 과학적 지식을 필요로 하지는 않지만, 실생활에서 접할 수 있는 어려울 것 같은 상황이나 문제들이 주어지고, 그러한 문제들을 해결하는 창의적인 방법들은 수학적, 과학적, 공학적으로 매우 효과적으로 활용될 수 있는 매우 큰 가치를 가지고 있다.

제시되는 문제와 상황들은 자신이 생각하지 못한 것들을 생각하게 하고, 복잡하고 어려울 것 같은 문제들을 빠르고 단순하게 해결할 수 있는 아이디어들을 살펴볼 수 있다.

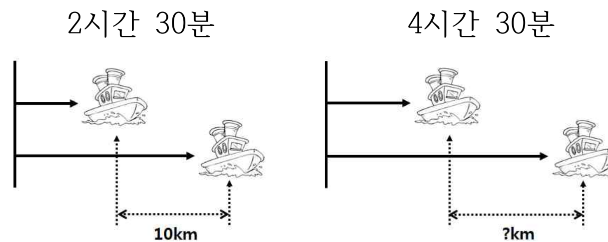
"문제해결의 도구!? 구체적 이해, 다양한 생각과 아이디어, 논리적 사고"

"나의 해결방법이 적당할까? 놓친 것은 없을까? 효과적일까?"

1. 계산->예측

가. 계산식 만들기

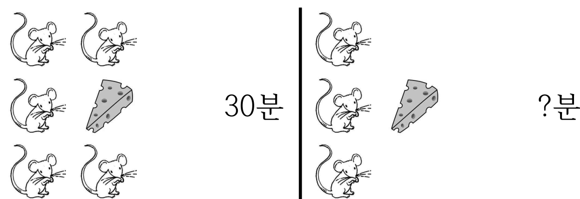
두 척의 배가 같은 위치에서 같은 방향으로 동시에 출발하여 일정한 속도로 나아간다. 2시간 30분 후 두 배 사이의 거리가 10km이었다. 이후 2시간을 더 항해한 후 두 배 사이의 거리는 몇 km 인가?



문제의 의미	- 제한된 정보만을 이용해 주어진 상황을 식으로 표현하고, 원하는 값을 찾아내기 위해 이용하거나 변형하는 연습
풀이 예시	- 주어진 내용 : 두 배의 이동 속도를 x, y라고 할 때, x배의 이동거리는 x*150분, y배의 이동거리는 y*150분. 따라서 2시간30분 후 두 배의 거리 차이는 x*150분-y*150분=10km. x-y=10km/150분 - 원하는 결과 : 4시간30분 후의 거리 차이는 x*270분-y*270분=(x-y)*270분=10km/150분*270분=18km

나. 처리량 계산하기(같은 성능)

5마리의 쥐가 함께 한 덩이 치즈를 먹는데 30분이 걸린다. 그렇다면, 3마리의 쥐가 함께 한 덩이의 치즈를 먹는 데는 몇 분이 걸릴까?

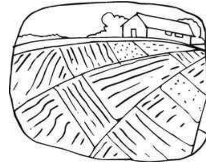


문제의 의미	- 똑 같은 장치 여러 개를 이용해 하나의 작업을 동시에 나누어 처리할 때 걸리는 처리 시간 예상. 멀티코어, 멀티프로세싱 등. 단 작업이 계산의 연관 관계가 없이 나누어질 수 있어야 함.
풀이 예시	- 주어진 내용 : 치즈1개/5마리=30분. - 원하는 결과 : 치즈1개/3마리=?분. 치즈1개/1마리=150분. 따라서 치즈1개/3마리=50분.

다. 처리량 계산하기(다른 성능)

어떤 밭을 일구는데, 세 사람 A, B, C가 혼자 밭을 일구면 각각 2시간, 3시간, 6시간이 걸린다. A, B, C 세 사람이 함께 이 밭을 일구면 몇 시간이 걸리는가?


"혼자하면 2시간"




"혼자하면 3시간"


"혼자하면 6시간"

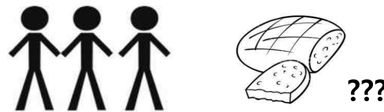
"3명이 동시에 함께 일하면 ?시간"

문제의 의미	- 서로 다른 장치 여러 개를 이용해 하나의 작업을 동시에 나누어 처리할 때 걸리는 처리 시간 계산. 분산 컴퓨팅 등. 단 작업이 계산의 연관 관계가 없이 나누어질 수 있어야 함.
풀이 예시	- 주어진 내용 : A능력=어떤밭/2시간, B능력=어떤밭/3시간, C능력=어떤밭/6시간. - 원하는 결과 : 어떤밭/(A능력+B능력+C능력)=?시간. 어떤밭/(어떤밭/2시간+어떤밭/3시간+어떤밭/6시간)=1/(1/2+1/3+1/6)=1시간.

2. 논리적 추론

가. 논리적으로 생각하기

갑, 을, 병 세 사람 중 한 사람만이 빵을 먹었다. 아래 세 사람의 말 중 두 사람의 말은 진실이고, 한 사람의 말은 거짓이라고 할 때, 거짓말을 한 사람과 빵을 먹은 사람은?



갑 : “내가 빵을 먹었다.”
을 : “갑의 말은 거짓말이다.”
병 : “을이 빵을 먹었다.”

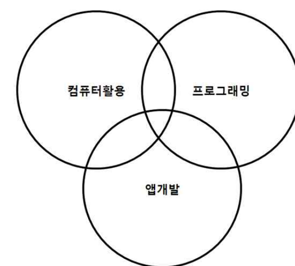
문제의 의미	- 주어진 상황을 이용해 논리적으로 원하는 결과를 이끌어내는 논리적, 연역적 사고 연습. 더 적은 조건을 이용해 효과적으로 원하는 결과를 얻어내는 연습
풀이 예시	- 주어진 내용 : 세 사람 중 한 사람만 빵을 먹었다. 세 사람 중 한 사람만 거짓이다. - 원하는 결과 : 갑이 거짓이라면, 을의 말이 참이 될 수 있고, 병의 말도 참이 될 수 있다. 따라서 갑이 거짓말을 했고, 을이 빵을 먹었다.

나. 집합과 논리연산 이해하기

철수네 반의 각 학생은 방과 후 수업으로 컴퓨터활용, 프로그래밍, 앱개발의 3과목 중 적어도 하나 이상을 들어야 한다. 이들 과목들의 수업 시간은 서로 다르므로 각 학생들은 여러 과목을 들을 수 있다. 이들 과목에 대하여 수업을 듣는 학생들의 수를 조사하여 다음과 같은 결과를 얻었다.

철수네 반 학생들은 모두 몇 명인가?

컴퓨터활용 수업을 듣는 학생 수 : 15명
프로그래밍 수업을 듣는 학생 수 : 20명
앱개발 수업을 듣는 학생 수 : 10명
컴퓨터활용과 프로그래밍 수업을 듣는 학생 수 : 9명
컴퓨터활용과 앱개발 수업을 듣는 학생 수 : 3명
프로그래밍 수업과 앱개발 수업을 듣는 학생 수 : 5명
컴퓨터활용, 프로그래밍, 앱개발 수업을 모두 듣는 학생 수 : 2명

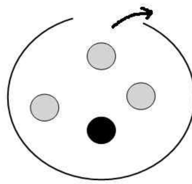


문제의 의미	- 주어진 내용을 집합을 이용해 표현하고, 원하는 내용을 얻어내기 위해 수정, 변형, 가공하는 연습. 집합, 논리연산, 벤다이어그램은 형태만 다르고 모두 정확히 같은 내용임. 예를 들어 논리연산 A+B는 A∪B와 완전히 같음.
풀이 예시	- 주어진 내용 : 모든 학생은 하나 이상의 과목을 수강, n(A)=15, n(B)=20, n(C)=10, n(A.B)=9, n(A.C)=3, n(B.C)=5, n(A.B.C)=2. 원하는 결과 : n(A+B+C)=?. 영역별로 합하면서 누적하고 원하는 부분을 만들기 위해 빼 나가면 아주 간단해진다.

3. 다양한 사고

가. 최대/최소 순열 생각하기

주머니 안에 빨간색, 파란색, 노란색 공이 각각 10개씩 들어 있다. 눈을 감고, 하나를 꺼낸 뒤 살펴보는 행동을 반복하다가 같은 색 공이 6개가 되는 순간 멈춘다. 최대 몇 개의 공을 꺼낼 수 있을까?



문제의 의미	- 여러 가지 가능성과 조합에 대한 다양한 생각을 이끌어내는 연습
풀이 예시	- 주어진 내용 : R, B, Y 각 10개 - 원하는 결과 : 같은 색 공이 6개가 되었을 때 최대로 꺼낼 수 있는 공의 개수는? 각각 5개 까지만 꺼내면 총합 15개, 이후 1개를 더 꺼내면 무조건 멈춤. 따라서 최대 16개

나. 최대/최소 집합 생각하기

찬호의 학교에는 100명의 학생이 있는데, 그 중 60명은 남자이고, 40명은 여자이다. 또 70명은 안경을 썼고, 30명은 안경을 쓰지 않았다.

그렇다면 안경을 쓴 여학생은 최소 몇 명인가?

문제의 의미	- 최대/최소 가능성에 대한 사고 연습
풀이 예시	- 주어진 내용 : 총 학생 수 100명, 60명은 남자, 40명은 여자. 70명은 안경착용. - 원하는 결과 : 안경 쓴 여학생으로 가능한 최소 인원은? 남학생이 모두 안경을 착용한다고 가정하면 10명

다. 여러 경우 생각하기

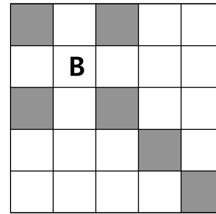
서양장기 체스에는 대각선 방향으로 움직일 수 있는 비숍(bishop)이 있다. 다음 그림과 같이 비숍이 B가 표시된 곳에 있을 때 비숍은 색칠된 칸에 있는 다른 말을 잡을 수 있다.

5×5 체스판에 비숍을 놓는데 서로가 서로를 잡을 수 없도록 하면서 최대한 많은

비숍을 놓고자 한다. 최대 몇 개의 비숍을 놓을 수 있을까?



주교(성직자)



<table>
<tr><td>문제의
의미</td><td>- 가능한 여러 가지 경우에서 최대/최소 탐색. 2차원 형태. 다른 말이나 장기 등으로 변형 가능.</td></tr>
<tr><td>풀이
예시</td><td>- 주어진 내용 : 한 비숍의 대각선으로는 다른 비숍을 놓을 수 없음.
- 원하는 결과 : 5*5 영역에 놓을 수 있는 비숍의 최대 개수는?
<table><tr><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>B</td><td>B</td><td>B</td><td></td></tr></table>등</td></tr>
</table>

라. 가능한 경우 만들기

각각 4리터와 7리터를 담을 수 있는 물통이 있다. 눈금이 없는 이 두 물통을 이용해서 원하는 양만큼의 물을 얻으려고 한다.



다음 물의 양 중, 두 물통만을 사용해서 잴 수 있는 양의 개수는 몇 개인가?

1리터, 2리터, 3리터, 5리터, 6리터

<table>
<tr><td>문제의
의미</td><td>- 가능한 여러 가지 경우 생각. 만들어진 결과로 다른 새로운 결과를 만들어 내는 생각이 중요.</td></tr>
<tr><td>풀이
예시</td><td>- 주어진 내용 : 4리터, 7리터 물통 1개씩.
- 원하는 결과 : 1/2/3/5/6 리터 중 가능한 개수는? 1리터(4->7, 4->7), 3리터(7->4), 2리터(1리터->7, 4->7, 4->7), 5리터(2리터->4, 7->4), 6리터(3리터->4, 7->4)</td></tr>
</table>

4. 관찰->분석->논리적 추론

가. 패턴 찾기

다음은 일정한 규칙에 따라 수를 늘어놓은 것이다. 빈칸에 가장 알맞은 수는?

1, 3, 6, 11, 19, 31, 48, ( ), ...

문제의 의미	- 임의 수열의 특성/패턴 찾아내고 정의/점화식 만들기
풀이 예시	- 주어진 내용 : 1, 3, 6, 11, 19, 31, 48, ... - 원하는 결과 : 1, 3, 6, 11, 19, 31, 48, (?), (??), ... 수 사이의 차이 2, 3, 5, 8, 12, 17 그 사이의 차이 1, 2, 3, 4, 5

나. 숫자 맞추기

A, B, C는 각각 1~9까지 숫자 중에 하나이고, 다른 알파벳은 다른 숫자를 나타낸다. 아래의 식을 만족하는 A의 값은?

```
    A  B  C
+   A  B  C
-----------
    B  B  A
```

문제의 의미	- 주어진 상황을 분석해서 문제 해결을 위한 특이점을 찾아내고 그 특성을 이용해 논리적으로 추론
풀이 예시	- 주어진 내용 : A, B, C는 각각 다른 숫자로 1~9 중 하나. 식 만족해야함. - 원하는 결과 : A는 어떤 수? 같은 수 C 2개를 합해 A가 나오니 A는 무조건 짝수. A가 6 또는 8이라면 100의 자리에서 자리올림수가 발생하게 되므로 A는 2 또는 4만 가능. A가 2라면 C는 1 또는 6. A:2, C:1 이면 B는 4 또는 5가 되어야 하지만 식을 만족하지 않음. A:2, C:6 이면 10의 자리 계산에서 B+B+1=10+B 가 되어야 하지만 100의 자리에서 불가능. A가 4라면 C는 2 또는 7. A:4, C:2 이면 10의 자리 계산에서 불가능. A:4, C:7 이면 B:9로 가능.

다. 특성 찾아 적용하기

1번부터 100번까지 번호가 붙여진 사람들이 있다. 또한 1부터 200번까지 번호가

붙여진 사물함이 있다. 처음에 모든 사물함의 문은 닫혀 있다. 번호 1번 사람부터 100번 사람까지 100명의 사람들이 모두 다음과 같은 작업을 수행 한다.



자기 번호의 배수가 되는 모든 번호의 사물함에 대하여 사물함이 닫혀있으면 열고, 열려있으면 닫는다. 1번부터 100번까지 모두 작업한 후, 마지막에 열려있는 사물함은 모두 몇 개인가?

문제의 의미	- 반복 작업 상황에서 특성을 찾아내 단계적으로 적용하고 최종결과를 찾아내는 연습
풀이 예시	- 주어진 내용 : 1~100번 학생, 1~200번의 사물함, 학생들은 자기 번호의 배수 사물함을 건드림. 열려있었으면 닫고, 닫혀있었으면 열음. - 원하는 결과 : 1~100번 학생이 모두 200번까지 사물함까지 작업을 반복하면 가장 나중에 열려있는 사물함은? 1~100번까지 1~100번 사물함에 작업하면 사물함 번호의 약수의 개수가 짝수이면 닫혀있고, 홀수이면 열려있게 됨. 약수의 개수가 홀 수개인 번호는 1, 4, 9, 16, 25와 같은 완전 제곱수. 따라서 1~100까지의 사물함 중에서 1, 4, 9, 16, 25, 36, 49, 64, 81, 100 은 열려있음. 1~100번까지 101~200번 사물함에 작업하면, 사물함 번호로는 다시 한 번 나눌 수 없기 때문에 완전 제곱수로 이루어진 사물함만 열려있게 됨. 121, 144, 169, 196 만 닫혀있게 됨.

5. 단계적 추론->계산

가. 반복 패턴 찾기

1을 7로 나누었을 때 소수점 이하 18번째 자리 수는?

문제의 의미	- 반복 계산의 패턴을 찾아 예상하는 연습, 패턴으로 계산 없이 출력가능
풀이 예시	- 주어진 내용 : 1/7의 소수점 계산(무한) - 원하는 결과 : 소수점 이하 18번째 자리수? 0.142857... 계속 반복. 18번째 자리수는 6번째 반복 수.

나. 연산자 이해하기

A◎B 는 A를 B로 나눈 몫이다.
(A◎4)◎5=3 일 때, A가 될 수 없는 자연수는?

문제의 의미	- 정의된 연산자의 기능 이해, 연산자의 복합사용 이해
풀이 예시	- 주어진 내용 : A◎B 는 A를 B로 나눈 몫 - 원하는 결과 : A로 가능한 자연수? (A◎4)◎5=3에서 A◎4 는 15~19까지 가능. A◎4==15 일 때, A는 60부터 63까지 가능... A◎4==19라면 A는 79까지 가능.

다. 수 개수 세기

1부터 2000까지의 정수들 중에서 4의 배수이고 6의 배수이면서 5의 배수가 아닌 수는 몇 개인가?

문제의 의미	- 조건에 맞는 수의 개수를 단계적으로 계산, 논리식으로 프로그래밍 가능
풀이 예시	- 주어진 내용 : 1~2000까지의 정수들. - 원하는 결과 : 4와 6의 배수이면서 5의 배수가 아닌 수의 개수? 4와 6의 배수는 최소공배수 12의 배수. 1~2000까지 12의 배수의 개수는 2000/12=166개. 12와 5의 배수는 60의 배수. 1~2000까지 60의 배수의 개수는 2000/60=33개. 따라서 133개.

라. 수 조합 계산하기1

약수가 2개인 수, 즉 1과 자신만을 약수로 갖는 수를 소수라고 한다. 8을 3개의 소수의 합으로 표현하는 방법은 2+3+3 한 가지 뿐이다.
28을 세 개의 소수의 합으로 나타내려고 한다면 가능한 경우는 모두 몇 가지인가?

문제의 의미	- 가능한 수 조합을 만들어 단계적으로 계산, 프로그래밍으로 검증 가능
풀이 예시	- 주어진 내용 : 28 - 원하는 결과 : 3개의 소수로 28을 만들어내는 모든 방법의 개수? 28보다 작은 모든 소수 : 2, 3, 5, 7, 11, 13, 17, 19, 23 3개의 소수 중 가장 작은 수가 2로 가능한 경우 : (2, 3, 23), (2, 7, 19), (2, 13, 13) 가장 작은 소수가 3, 5, 7, 11, 13, 17, 19, 23인 경우 불가능. **참고: 골드바흐 추측(2보다 큰 모든 짝수는 소수 2개의 합으로 만들 수 있을 것이다?)

마. 수 조합 계산하기2

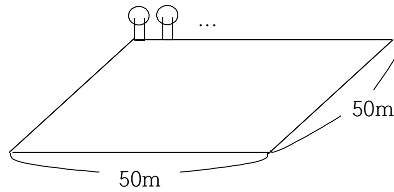
기약분수는 분자와 분모의 최대공약수가 1인 분수이다.
분수 $\frac{7}{12}$와 분수 $\frac{13}{16}$사이의 분수 중에서 분모가 24인 기약분수는 몇 개인가?

$$\frac{7}{12} \leq \quad \cdots \quad \leq \frac{13}{16}$$

문제의 의미	- 가능한 수 조합을 만들어 단계적으로 계산, 프로그래밍으로 검증 가능
풀이 예시	- 주어진 내용 : 7/12이상 13/16이하 사이에 있는 분모가 24이고 분자와 공약수가 1뿐인 분수(기약분수) - 원하는 결과 : 분모가 24이면서 24와 공약수가 1뿐인 경우의 수? 최소 7/12(14/24), 최대13/16(19.5/24) 인 분수이므로, 14 ~ 19.5의 정수 중에서 24와 약분되지 않는 정수는 17, 19 뿐

바. 자원 계산하기

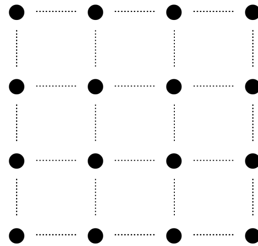
너비가 50m인 정사각형 모양의 땅 경계에 10m 간격으로 나무를 심으려고 한다.
총 몇 그루의 나무가 필요한가?



문제의 의미	- 주어진 문제를 정확히 이해하고 단계적으로 해결하는 연습
풀이 예시	- 주어진 내용 : 50*50 크기의 땅. 둘레에만 10간격으로 나무 심기 - 원하는 결과 : 몇 그루 필요? 50길이 한 변에 6그루 가능. 4변이지만 면적으로 모두 채우는 것이 아니므로 중복 고려하면 한 번에 5그루씩 4변으로 계산하면 총 20그루 가능. 문제의 정확한 이해와 중복 제거가 중요.

사. 도형 개수 세기

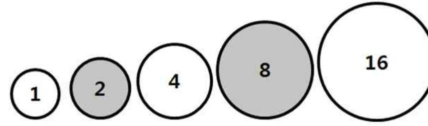
16개의 점이 아래와 같이 놓여 있다. 점선으로 이은 두 점사이의 거리가 모두 같다고 할 때, 서로 다른 네 점을 이어 만들 수 있는 정사각형의 개수는 모두 몇 개인가?



문제의 의미	- 가능한 모든 경우를 단계적으로 찾아 해결하는 연습
풀이 예시	- 주어진 내용 : 4*4 정사각형 점. - 원하는 결과 : 그 중 4개의 점을 선택해 정사각형으로 만들 수 있는 경우의 개수? 단계적 사고가 매우 중요. 한 변의 길이가 1인 경우 : 9가지 한 변의 길이가 $\sqrt{2}$인 경우 : 4가지 한 변의 길이가 2인 경우 : 4가지 한 변의 길이가 $\sqrt{5}$인 경우 2가지 한 변의 길이가 3인 경우 : 1가지

아. 가능한 수 계산하기

5개의 동전이 있다 : 1원짜리 동전 하나, 2원짜리 동전 하나, 4원짜리 동전 하나, 8원짜리 동전 하나, 16원짜리 동전 하나.



위의 동전들을 사용하여 만들 수 있는 있는 서로 다른 금액은 모두 몇 가지인가? (단, 한 개 이상의 동전을 사용하여야 한다.)

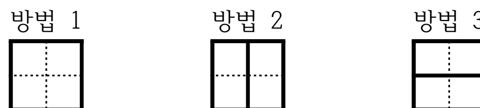
문제의 의미	- 가능한 모든 경우를 단계적으로 찾아 해결하는 연습
풀이 예시	- 주어진 내용 : 1, 2, 4, 8, 16 동전 각 1개 - 원하는 결과 : 1개 이상의 동전으로 만들 수 있는 서로 다른 금액의 수? 2진수 진법체계와 동일. 1~31까지 가능.

자. 가능한 종류 계산하기

다음 그림과 같은 2가지 종류의 블록이 있다.

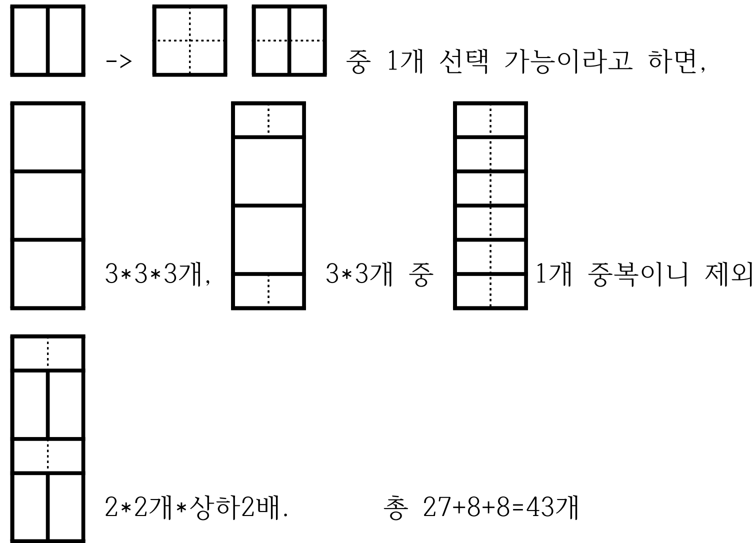


위의 블록들을 여러 개 사용해서 가로 2칸, 세로 2칸 크기의 판을 채우는 방법은 다음 3가지이다. 이 때, 블록들을 회전해서 채워 넣는 것도 가능하다.



그렇다면, 위의 2가지 블록들을 여러 개 사용해서 가로 2칸, 세로 6칸 크기의 판을 채우는 방법은 모두 몇 가지인가?

문제의 의미	- 가능한 모든 경우를 단계적으로 찾아 해결하는 연습, 공학적 응용
풀이 예시1	- 주어진 내용 : 2 종류의 블록, 조합방법 3개 - 원하는 결과 :가로2*세로6크기로 만들 수 있는 서로 다른 모양의 블록의 개수? -> 중 1개 선택 가능



풀이 예시2

귀납적인 방법으로 해결하기

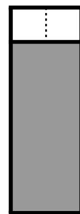
1) : 기본 블록인 1*2모양의 블록을 채우는 경우의 수는 1가지

2) : 2*2블럭을 채우는 경우의 수는 3가지

1), 2)를 이용해서 3이상의 n에 대해서 n*2를 채우는 경우를 귀납적으로 생각할 수 있다.

n*2를 채우는 경우의 수를 구해보자. 귀납적으로 구하기 위해서는 마지막에 채우는 방법에 집중해야 한다. 다음의 경우를 살펴보자.

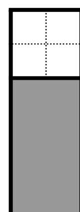
3)



n*2판 중 (n-1)*2판이 이미 채워져 있다면 마지막에 를 통하여 전체를 채울 수 있다. 즉 마지막에 를 통해 채우는 경우의 수는 (n-1)*2판을 채우는 경우의 수와 같다

다음으로 3)과 중복되지 않으며 마지막 2행을 채우는 방법을 생각해보자.

4)



이번에는 (n-2)*2판이 채워져 있다. 이 상태에서 를 이용해서 바로 채울 수 있으며, 를 이용해서도 바로 채울 수 있다. 하

지만 [그림] 를 이용하는 것은 위의 3)에서 채운 경우와 중복되므로 배제하자. 3)과 중복되지 않으면서 4)의 상태에서 n*2판을 채우는 경우의 수는 (n-2)*2판을 채운 후 2가지 방법이 있으므로 {(n-2)*2의 경우의수}*2가지가 된다.

3행 이상 채우는 경우는 모두 3), 4)와 중복되므로 구할 필요가 없다. 따라서, n*2판을 채우는 경우의 수를 f(n)이라고 하면 다음과 같은 식을 얻을 수 있다.

f(n) = f(n-1) + 2*f(n-2)와 같다. (단, n>2, f(1)=1, f(2)=3) ... 식①

위 식①과 아래의 표를 이용하면 구하고자하는 f(6)을 쉽게 구할 수 있다.

일단 f(1), f(2)는 직접 구할 수 있으므로 표는 다음과 같다.

f(1)	f(2)	f(3)	f(4)	f(5)	f(6)
1	3				

나머지 부분은 식①을 이용해서 채우면 원하는 답을 구할 수 있다.

f(1)	f(2)	f(3)	f(4)	f(5)	f(6)
1	3	5	11	21	43

따라서 43가지. 위 표를 잘 이용하면 6보다 훨씬 큰 숫자에 대해서도 쉽게 구할 수 있다.

6. 단계적 추론->계산->최적해

가. 수 조합하기

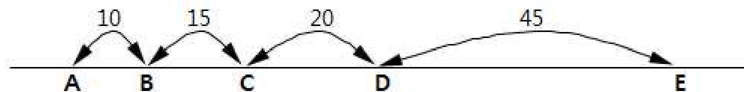
두 자연수의 곱이 1000일 때, 가능한 두 자연수 합의 최솟값은?

문제의 의미	- 가능한 모든 경우를 단계적으로 찾아 최적해를 계산하는 연습
풀이 예시	- 주어진 내용 : 두 자연수의 곱으로 1000 만들기 - 원하는 결과 : 두 자연수 합의 최솟값? 작은 수부터 선택한 가능한 모든 경우 (1, 1000), (2, 500), (4, 250), (5, 200), (10, 100), (20, 50), (25, 40)

나. 최적 위치 계산하기

다음 그림과 같이 A, B, C, D, E 다섯 학생의 집이 일직선상에 위치해 있으며, 집 사이의 거리는 그림과 같다.

어느 날 다섯 명의 학생이 그 중 한 학생의 집에서 모이기로 하였을 때, 누구의 집에서 모여야 모든 학생이 이동한 거리의 합이 최소가 될까?

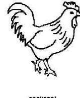


문제의 의미	- 가능한 모든 경우를 단계적으로 찾아 최적해를 계산하는 연습, 계산기하
풀이 예시	- 주어진 내용 : 위치와 각각 사이의 거리 - 원하는 결과 : 모두가 이동했을 때 최소 합의 위치? A:90, B:140, C:135, D:145, E:280

다. 최적 조합 찾아내기1

암탉은 한 마리에 15원, 수탉은 한 마리에 9원, 병아리는 한 마리에 1원이다. 300원을 모두 사용하여 정확히 100마리를 사는데 암탉을 되도록 많이 사려고 한다면 수탉은 최대 몇 마리 살 수 있을까?





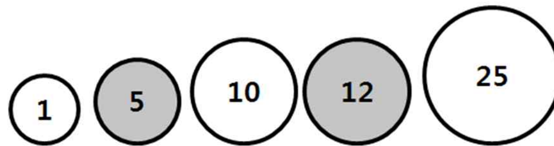


15원 9원 1원

문제의 의미	- 가능한 모든 경우를 단계적으로 찾아 최적 조합을 계산하는 연습
풀이 예시	- 주어진 내용 : 종류와 가격, 300원, 100마리 - 원하는 결과 : 암탉을 최대로 구입할 때, 수탉은 최대 몇 마리 구입가능? 각 A, B, C마리라고 할 때, 15A+9B+C=300, A+B+C=100, A/B/C 정수 가능한 A를 먼저 최대로 채우고 나머지로 조건 확인해서 가능 조합 찾기 A=14, 13이면 불가능. A=12, B=4, C=84로 가능

라. 최적 조합 찾아내기2

A나라에는 1원, 5원, 10원, 12원, 25원의 다섯 가지 동전이 있다. 254원을 가장 적은 개수의 동전으로 가지고자 한다면 25원짜리가 몇 개가 될까?



문제의 의미	- 가능한 모든 경우를 단계적으로 찾아 최적 조합을 계산하는 연습, 제일 큰 가치를 우선적으로 선택했을 때(그리디) 안 되는 경우.
풀이 예시	- 주어진 내용 : 종류와 가격, 254원 - 원하는 결과 : 254원을 만들 수 있는 최소 동전 개수? 25, 12, 10, 5, 1 동전의 개수를 (0, 0, 0, 0, 0)로 표현했을 때 가능한 경우.. (10, 0, 0, 0, 4) 합 14개, (9, 2, 0, 1, 0) 합 12개, (8, 4, 0, 1, 1) 합 14개... 순서대로 보다 큰 동전을 하나씩 줄여가면서 순차적으로 계산하면 됨.

마. 최대 가치 찾기

다음은 물건 A, B, C, D, E의 무게(단위 Kg)와 가치(단위 만원)이다.

물 건	A	B	C	D	E
무 게	3	4	7	8	9
가 치	4	5	10	11	13



23Kg?

철수가 가방에 23Kg의 무게를 넣어 가질 수 있다면, 철수가 얻을 수 있는 최대 가치는 얼마인가?(단, A, B, C, D, E는 모두 여러 개씩 있다.)

문제의 의미	- 가능한 모든 경우를 단계적으로 찾아 최적 조합을 계산하는 연습, 제한된 자원으로 최대 이익 찾기(배낭문제, knapsack-냅색-)
풀이 예시	- 주어진 내용 : 종류/무게와 가격, 23kg - 원하는 결과 : 23kg으로 얻을 수 있는 최대 가치? 9, 8, 7, 4, 3 무게의 개수를 (0, 0, 0, 0, 0)으로 표현했을 때 가능한 경우 (1, 1, 0, 0, 2) 가치:32, (1, 0, 2, 0, 0) 가치:33, (1, 0, 1, 1, 1) 가치:32, 같은 방법으로 (1, 0, 0, 2, 2), (0, 2, 1, 1, 1), (0, 2, 0, 1, 1), (0, 1, 0, 3, 1)..?

7. 문제해결 전략 개발

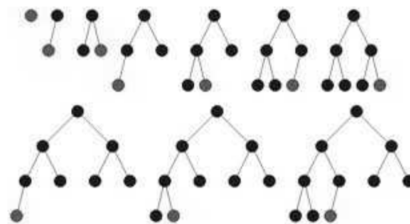
가. 구조 찾아내기

다음과 같이 번식을 하는 동물이 있다.

1) 태어나서 정확히 1년 만에 2마리 이하의 새끼를 낳는다.

2) 태어나서 1년이 지난 이후에는 더 이상 새끼를 낳지 않는다.

갓 태어난 새끼 한 마리로부터 번식된 동물이 이 새끼를 포함하여 모두 270마리였다. 이와 같이 되는데 걸린 기간의 최소 년 수는 얼마인가?



문제의 의미	- 주어진 상황을 이해하기 편리한 구조로 표현하는 연습, 2진 트리의 특성
풀이 예시	- 주어진 내용 : 한 마리->1년 뒤 2마리로 번식하고 번식 중지, 계속 반복 - 원하는 결과 : 270마리가 되는 최소 년 수? full binary tree 구조로 그려보면 각 레벨은 필요한 년 수를 나타내고, 그 때까지 최대 마리수가 노드로 표현됨. 각 레벨에서 노드의 개수는 1, 2, 4, 8, 16, 32, 64, 128, 256 ... 으로 계속 증가. 그 합이 270을 넘어갈 때는 최소 8년이 지나면 됨. 공식보다는 구조 자체만으로 쉽게 해결 가능.

나. 계산 누적하기

아래 그림은 숫자 삼각형을 보여준다. 맨 꼭대기에서 바닥에까지 한 층에 하나씩 연결되는 길을 찾아 내려가는데 그 합이 최대가 되는 것을 구하려는 문제이다.

여기서 한 층씩 내려간다는 것은 대각선 방향으로 왼쪽 혹은 대각선 방향으로 오른쪽으로 내려가는 것이다. 최대가 되는 합은 얼마인가?

7

3 8

8 1 4

2 6 5 4

3 6 2 7 1

7 7 6 5 5 7

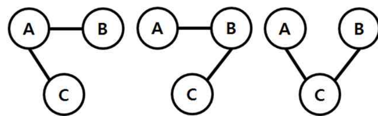
문제의 의미	- 주어진 상황을 정확히 해결하기 위한 해결방법 설계 연습, 동적 계획
풀이 예시	- 주어진 내용 : 숫자 선택 방법 - 원하는 결과 : 가장 아래까지 선택했을 때 최대 합의 경우는? 갈림길에서 가장 좋은 다음 값으로 선택해 가면 하면 전체적으로 봤을 때 최대가 아닌 경우가 발생함. 이전의 값 중에서 가장 좋은 값을 선택해 누적하는 것이 정확한 답을 얻을 수 있음. 욕심쟁이(그리디)기법으로 정확한 답을 얻지 못할 수 있는 예시. 이전의 값을 참조해서 가장 최적의 결과를 계산해 최적해를 쌓아가는 동적계획법으로 정확한 답을 얻는 예시.

다. 연결 만들기

도시 공사에서는 A, B, C, D, E의 5개 아파트 단지를 연결하는 도로를 놓으려고 한다. 여기서 A단지와 B단지가 연결되고 또 B단지와 C단지가 연결되면 A단지와 C단지도 연결된 것이다.

각 단지를 직접 연결하는데 드는 비용이 아래 표와 같다면, 5개 단지를 연결하는데 드는 가장 적은 비용은 얼마인가?

연결된 A, B, C 단지의 예시



	A	B	C	D	E
A		8	12	6	19
B	8		18	16	7
C	12	18		17	11
D	6	16	17		10
E	19	7	11	10	

문제의 의미	- 주어진 상황을 정확히 해결하기 위한 구조/알고리즘 설계 연습. 최소비용스패닝 트리(네트워크 설계, 도시 설계 등에 응용)
풀이 예시	- 주어진 내용 : 노드와 연결 관계/가중치를 의미하는, 2차원 배열로 표현된 인접 행렬 - 원하는 결과 : 모든 도시들로 이동할 수 있도록 최소의 비용을 들여 연결을 만들면? 도시들의 연결 관계를 노드와 간선/가중치를 이용해 그래프로 표현한 후, 가장 적은 연결비용이 드는 간선을 순서대로 선택해서 모든 도시들이 연결되는지 확인 함. 단, 간선 연결시 사이클이 발생하면 선택하지 않음.

라. 표 해석하기

키가 모두 다른 10명의 학생들 A, B, C, D, E, F, G, H, I, J가 순서대로 줄을 서 있다.(A가 가장 앞에 있음). 각자 뒤에 있는 사람들 중에서 자기 자신보다 키가 작은 사람의 수를 세어 보았더니 아래 표와 같았다.

E보다 키가 큰 사람은 몇 명인가?



A	B	C	D	E	F	G	H	I	J
4	5	0	6	3	2	3	1	1	0

문제의 의미	- 테이블(배열)로 주어지는 상황을 이용해 답을 찾아내는 연습. 테이블의 해석과 문제해결을 위한 효과적인 아이디어 산출이 핵심.
풀이 예시	- 주어진 내용 : 자신의 뒤에 서 있는 사람들 중에서 자기보다 키가 작은 사람들의 수가 작성되어있는 표 - 원하는 결과 : 키 순서? J가 마지막에 있으니 뒤를 보니 자기보다 작은 사람은 뒤에 없음. I가 뒤를 보니 J가 있고 작은 사람이 1명이니 I는 J보다 큼. H가 뒤를 보니 J, I가 있는데 자기보다 작은 사람이 1명뿐 이니 키는 I>H>J 로 해석할 수 있음. 계속적으로 왼쪽으로 가면서 데이터를 확인하고 키순서로 배열하면 쉽게 문제 해결 가능.

코드 이해

대회의 문제들은 가장 기초적인 형태의 간단한 상황들로 주어지고, 문제들을 해결하기 위한 방법들은 무수히 많다.

손과 연필로 계산할 수 없을 정도로 복잡해지는 경우에는, 보다 빠른 시간 내에 원하는 결과를 얻어내기 위해 핵심적인 알고리즘을 프로그래밍을 통해 실제로 구현해 컴퓨터의 계산 수행능력을 사용하고, 아주 빠르게 원하는 결과를 얻어낼 수 있다. 하지만 정보과학의 프로그래밍은 단순히 컴퓨터라는 빠른 기계를 동작시키기 위해 프로그래밍언어를 활용하는 것과는 다른 의미를 가진다.

일반적인 프로그래밍이 단순히 기계를 작동시키기 위한 방법으로의 관점이 주요 내용이라면, 문제해결에서 바라보는 프로그래밍은 제한적인 기계의 계산 수행능력 안에서 최대의 효율 즉, 더 빠른 수행능력과 더 짧은 응답시간을 얻어내기 위한 사람들의 창의적인 생각과 아이디어, 도전이 주요 내용이 된다.

보다 빠르게, 보다 효율적으로 원하는 결과를 얻어낼 수 있도록 하는 창의적인 아이디어들은 현재의 고성능 컴퓨터의 설계 뿐 아니라, 매우 제한적 성능의 하드웨어를 가진 스마트 기기에서 아주 빠른 소프트웨어들을 만들어낼 수 있는 핵심 원리가 된다. 2배 더 빠른 하드웨어를 개발하는 것이 어려울까? 2배 더 빠른 아이디어와 알고리즘을 생각하는 것이 어려울까?

“자! 여러분들은... 더 빠른 자동차를 만드시겠습니까?”

“더 빠른 길을 찾으시겠습니까?”

1. 반복구조 이해하기

가. 반복 따라가기1(±1 증감)

다음은 어떤 프로그램의 일부이다. 이 부분을 실행시킨 후 a의 값은?

```
t=0;                        //t에 0저장
for(i=10; i<=14; i++)  //i를 10부터 14까지 1씩 증가시키면서
    t += i;                 //t = t + i 반복
a = t/5;                    //t/5 는 t를 5로 나눈 몫
```

문제의 의미	- C언어 소스코드 이해하기 연습. for 단순 반복 구조의 이해, ±1 증감
풀이 예시	10부터 14까지 1씩 증가시키며 t에 저장. i: 10, 11, 12, 13, 14 t: 10, 21, 33, 46, 60 , 60/5 ** 도대체 i는 뭘까? iterator?

i	10	11	12	13	14
t	10	21	33	46	60

나. 반복 따라가기2(±n 증감)

다음은 어떤 프로그램의 일부이다. 이 부분을 실행시킨 후 c의 값은?

```
c=0;
for(i=1; i<=100; i+=7)        //i를 1부터 100까지 7씩 증가 반복
    c++;
```

문제의 의미	- C언어 소스코드 이해하기 연습. for 단순 반복 구조의 이해, ±n 증감
풀이 예시	1부터 100과 같거나 작을 때까지 7씩 증가시키며 t에 저장.

i	1	8	...	92	99
c	1	2	...	14	15

다. 반복 따라가기3(중첩)

다음은 어떤 프로그램의 일부이다. 이 부분을 실행시킨 후 k의 값은?

```
k=0;
for(i=1; i<=3; i++)       //i를 1부터 3까지 1씩 증가시키면서
    for(j=1; j<=i; j++) //j를 1부터 i까지 1씩 증가
        k+= i+j;          //k = k + i + j;
```

문제의 의미	- C언어 소스코드 이해하기 연습. for 중첩 반복 구조의 이해
풀이 예시	- i+j의 결과를 k에 누적

-

i	1	2	2	3	3	3
j	1	1	2	1	2	3
k	2	5	9	13	18	24

2. 조건식 이해하기

가. 조건식 사용하기

다음은 1부터 100까지 합을 구하는 프로그램의 일부이다. 빈 칸에 들어갈 내용으로 가장 알맞은 것은?

```
s=0;
c=1;
while(          )  //□의 조건인 동안 반복
{
    s=s+c;
    c++;
}
```

① c<99 ② c<100 ③ c==100 ④ c!=100 ⑤ c<101

문제의 의미	- C언어 소스코드 이해하기 연습. while(조건식) 반복 구조의 이해, 경계에서 주의하기
풀이 예시	- c를 1부터 100까지 1씩 증가시키면서 s에 누적. - c<99, c<100, c!=100 이면 100이 누적 되지 않음. c==0이면 실행 안 됨.

나. 논리식 만들기

년도가 4로 나누어지면 윤년이다. 그러나 년도가 100으로 나누어지면 평년이다. 그러나 년도가 400으로 나누어지면 윤년이다. 주어진 년도가 윤년인지 판단하는 함수를 다음과 같이 작성하려고 한다. 빈 칸에 들어갈 내용으로 알맞은 것은?

```
int f(int y)
{
  if(          ) return 0; //평년이다.
  else return 1;           //윤년이다.
}
```

① (y%4==0 && y%100!=0) && y%400==0
② (y%4==0 && y%100!=0) || y%400==0
③ (y%4!=0 && y%100==0) || y%400!=0
④ (y%4!=0 || y%100==0) || y%400!=0
⑤ (y%4!=0 || y%100==0) && y%400!=0

문제의 의미	- C언어 소스코드 논리식 만들기 연습. 주어진 상황을 논리식을 이용해 정확히 구성하거나, 다른 형태로 변환하거나 이해할 수 있는지
풀이 예시	- y를 계산해 윤년 계산. - 논리식은 집합연산/벤다이어그램과 정확히 같은 내용. 평년만 계산하면 됨, 말 그대로 표현하면 (y%4!=0)+(y%4==0 && y%100==0)-(y%400==0) 변형하는 것 보다 벤다이어그램으로 그리고 비교하면 간단해짐. [4 / 100 / 400 벤다이어그램] 과 같은 것을 나타내는 것은?

3. 실행 과정 이해하기(코드 이해하기)

가. 반복 중 변화 이해하기

다음은 어떤 프로그램의 일부이다. 이 부분을 실행시킨 후 c의 값은?

```
k=10;
c=0;
do                         //반복 구역 시작
{
    if(k%2==0) k/=2; //k가 짝수이면 나눈 몫을 저장
    else k=3*k+1;     //그렇지 않으면 3*k+1을 저장
    c++               //c의 값 1 증가
}while(k>1);          //k>1 인 동안 반복
```

문제의 의미	- C언어 소스코드 이해하기 연습. 반복 중 변환 따라가기
풀이 예시	- k가 짝수이면 반으로 나누고, 홀수이면 3배로 곱한 후 1더하기 - k: 10, 5, 16, 8, 4, 2, 1 / c: 0, 1, 2, 3, 4, 5, 6

k	10	5	16	8	4	2	1
c	0	1	2	3	4	5	6

나. 논리식 이해하기

다음은 어떤 프로그램의 일부이다. 이 부분을 실행시킨 후 c의 값은?

```
c=0;
for(i=1; i<=30; i++)       //i를 1부터 30까지 1씩 증가 반복
    if(i%3==0 && i%5==0) c++; //3의 배수이고 5의 배수이면 c++
```

문제의 의미	- C언어 소스코드 이해하기 연습. 논리식의 의미 파악하기
풀이 예시	- 3의 배수이면서 5의 배수인 수는 15의 배수, 따라서 15, 30

다. 조건식 예측하기

다음은 자연수 n이 소수인지 아닌지를 출력하는 프로그램의 일부이다. 빈 칸에 들어갈 내용으로 가장 알맞은 것은?

```
for(i=2; i<n; i++)          //i는 2부터 n-1까지 1씩 증가
    if(n%i==0) break;       //n을 i로 나눈 나머지가 0이면 정지

if(          ) printf("%d 소수\n", n); //□이면 소수 출력
else printf("%d 합성수\n", n);
```

① i==n ② i>n ③ i!=n ④ i<n ⑤ i<=n

문제의 의미	- C언어 소스코드 이해하기 연습. 계산 결과의 의미 파악하기
풀이 예시	- n을 2부터 n보다 작은 수로 계속 나누어 보다가 나누어 떨어지면 정지. - 소수라면? 주어진 범위에서 나누어 떨어지지 않고 최종 i의 값이 n과 같아지고 반복이 종료됨

라. 패턴 파악하기

다음은 어떤 프로그램의 일부이다. 이 부분을 실행시킨 후 t의 값은?

```
i=t=0;                  //t에 0저장 후 i에 t값 저장
while(i<30)             //t가 30보다 작은 동안 반복
{
    i++;                //i 1만큼 증가
    if(i%2==0) t+=i; //i가 짝수이면 t에 i값을 더함
    else t-=i;          //아니면 t에서 i값을 뺌
}
```

문제의 의미	- C언어 소스코드 이해하기 연습. 반복적인 계산의 형태 파악하기
풀이 예시	- 1부터 1씩 증가시키면서 짝수이면 더하고, 홀수이면 빼가면서 누적 패턴은 -1 +2 -3 +4 -5 +6 ...-29 + 30 2n-1만큼 진행하면 -n으로 합 계산, 2n만큼 진행하면 +n으로 합 계산

마. 수학적 알고리즘 이해하기

아래 프로그램의 실행 후 a의 값은?

```
a=30;
b=16;
while(b>0)      //b가 0보다 큰 동안 반복
{
    c=a%b;      //a를 b로 나눈 나머지를 c에 저장
    a=b;        //b의 값을 a에 저장
    b=c;        //c의 값을 b에 저장
}
```

문제의 의미	- C언어 소스코드 이해하기 연습. 수학 알고리즘 구현 이해하기
풀이 예시	- a를 b로 나눈 나머지 계산, a에 b값 저장, b에 그 나머지 저장 계속 반복해가다가 b가 0이 되면 a는 가장 처음 두 수의 최대 공약수가 됨. 유클리드 호제법(서로 건너뛰는 방법), 몇 가지 변형 가능, 한 쪽을 계속 빼가면서 더 이상 못 뺄 때까지 계속 반복하면 똑같은 작업이 됨. 서로 뺀 수를 이용해 계산을 할 때, 최대 공약수가 나오는 이유.. 어떤 두 수가 최대 공약수 k를 가지고 있다면, 그 두 수를 빼도 k의 배수가 되기 때문. a=16, b=30이라도 됨.

4. 배열 구조 이해하기(코드 이해하기)

가. 배열 채우기1(단순)

다음은 어떤 프로그램의 일부이다. 이 부분을 실행시킨 후 a[7]의 값은?

```
a[1]=1;                        //a[]= |  | 1 |  |  |  |  |  |  |  |  |
a[2]=1;                        //a[]= |  | 1 | 1 |  |  |  |  |  |  |  |
for(i=3; i<=10; i++)            //i를 3부터 10까지 1씩 증가, 반복
    a[i]=a[i-1]+a[i-2]+1;  //앞의 2개 수의 합 +1 해서 저장
```

문제의 의미	- 배열을 이용하는 계산의 누적. 동적 계획법과 연관
풀이 예시	- 처음에 1, 1을 저장하고 그 이후에는 이전 두수의 합에 1을 더해 계산

	0	1	2	3	4	5	6	7
a[] :		1	1	3	5	9	15	25

나. 배열 채우기2(의미)

다음은 어떤 프로그램의 일부이다. 이 부분을 실행시킨 후 a[100]의 값은?

```
a[1]=1;                        //a[]= |  | 1 |  |  |  |  |  |  |  |  |
for(i=2; i<=100; i++)           //i를 2부터 100까지 1씩 증가 반복
    a[i]=a[i/2]+1;
```

문제의 의미	- 배열을 이용한 값 채우기. 배열을 이용한 완전 2진 트리와 연관
풀이 예시	- 1번 공간을 2진 트리의 루트라고 생각하면, 2진 트리의 왼쪽 자식은 2번, 오른쪽 자식은 3번. 일반화 하면 n번 노드의 왼쪽 자식은 2n번, 오른쪽 자식은 2n+1번이라고 생각하고 저장할 수 있음.

a[1] 1 — 1개
a[2] 2, a[3] 2 — 2개
a[4] 3, a[5] 3, a[6] 3, a[7] 3 — 4개

100번에는? 1+2+4+...+n<=100

다. 배열 채우기3(2중 참조)

다음은 어떤 프로그램의 일부이다. 이 부분을 실행시킨 후 a[5]의 값은?

코드	주석							
a[1]=2;	//a[]=		2					
a[2]=3;	//a[]=		2	3				
a[3]=4;	//a[]=		2	3	4			
a[4]=5;	//a[]=		2	3	4	5		
a[5]=1;	//a[]=		2	3	4	5	1	

```
for(i=1; i<=5; i++)          //i를 2부터 5까지 1씩 증가 반복
    a[i]=a[a[i]];
```

문제의 의미	- 배열의 2중 참조를 활용한 데이터의 이동과 결과 파악하기
풀이 예시	- 오른쪽 값이 왼쪽으로 이동해가는 형태. 마지막 단계 주의!

5. 함수 실행 이해하기(코드 이해하기)

가. 조건에 따라 처리하기

다음과 같은 함수 f가 구현되어 있을 때 f(10, f(3,7))의 값은?

```
int f(int a, int b)        //두 개의 정수를 a,b에 저장해 계산
{
    if(a > b) return a-b; //a>b 이면 a-b의 결과를 돌려줌
    else return b-a;       //아니면 b-a의 결과를 돌려줌
}
```

문제의 의미	- 함수의 동작 파악하기
풀이 예시	- 값을 2개 받아 자신이 사용하는 a와 b로 저장해 계산. 두 정수의 차를 계산해 돌려주는 함수 f(3, 7)은 3과7의 차이인 4로 바뀜. f(10, 4)는 6으로 계산

나. 재귀적으로 처리하기

다음과 같은 함수 f가 구현되어 있을 때 f(6)의 값은?

```
int f(int a)
{
    if(a<=2) return a;          //a<=2 이면 a값 돌려줌
    else return f(a-1)*f(a-2);//아니면 f(a-1)*f(a-2) 돌려줌
}
```

문제의 의미	- 재귀 함수의 동작 파악하기
풀이 예시	- 2이하인 수를 넘겨주면 그대로 계산 결과를 돌려주고, 2보다 큰 수를 넘겨주면 변수 a에 저장해 f(a-1)*f(a-2)를 계산해 돌려줌. a의 값이 2이하가 될 때까지 계산을 잠시 멈추고 반복적으로 재귀호출. 계산이 되면 잠시 멈춰두었던 계산을 다시 진행. - 2개의 함수를 다시 호출하므로 2진 트리 형식으로 전개됨(백트랙킹). 살펴보면, 똑같은 계산을 중복하는 경우가 많이 생김... - 따라서, f(6)을 계산하기 위해 역으로 a가 1, 2 일 때부터 계산해서 쌓아 올라가는 것이 더 간단(다이나믹)

a	1	2	3	4	5	6
f(a)	1	2	2	4	8	32

다. 재귀함수 설계하기

다음은 1부터 n까지의 합을 구하는 재귀함수를 구현한 것이다. 빈 칸에 알맞은 식은?

```
int f(int n)
{
    if(n==1) return 1;
    else return [          ];
}
```

① n+f(n)　② (n-1)+f(n)　③ n+f(n-1)　④ (n-1)+f(n-1)　⑤ 1+f(n-1)

문제의 의미	- 재귀 함수 설계 연습. 한 방향 재귀호출 이해하기. 이전과의 변화에 대해서만 정의해주면 함수가 재귀적으로 호출되면서 실행.
풀이 예시	- 1~n까지의 합은 1~(n-1) 까지의 합에 n만 더하면 됨. 주어진 어떤 수 n까지의 합을 만들어 주는 함수를 f(n)이라고 한다면, 일반식(점화식) f(n)=f(n-1)+n 으로 재귀적 함수 정의 가능. 단, 재귀 호출을 중지시키기 위한 임계조건(n==1)을 반드시 정의해 둠.

라. 재귀적 실행 이해하기(분할)

아래와 같은 함수를 작성하여 f(100)을 실행하였을 때, c의 값은?

```
int c;
void f(int n)          //정수 1개를 전달받아 n에 저장
{
    if(n<=1) c++;      //n<=1 이면 c에 저장된 값을 1만큼 증가
    else               //그렇지 않으면
    {
        f(n/2);        //n을 2로 나눈 몫을 f( ) 에 전달
        f(n/2);        //한 번 더 f(n/2) 실행
    }
}
```

문제의 의미	- 재귀 함수 설계 연습. 여러 방향 재귀호출 이해하기. 이전과의 변화에 대해서 순서대로 정의해주면 함수가 재귀적으로 분기(분할) 호출되면서 실행.
풀이 예시	- 전역 변수 c, 모든 함수가 접근해 참조/기록 가능. 재귀호출 함수. 중지 조건은 n<=1 인 경우. 아니면 n/2로 나뉘어 2개로 순차적으로 호출 됨. - 재귀호출의 분기 호출 방향은 깊이우선방식으로 계산 진행. - 1이 되면 전역 변수 c의 값을 1증가 시키고 재귀 호출 중단. - f(1)이 호출된 만큼 c의 값 증가 되므로 f(1) 노드의 개수가 c의 값이 됨. - 함수 f(100)의 재귀 호출을 2진 트리로 그렸을 때, 가장 아래 레벨에서 f(1) 노드의 개수를 계산하면 됨. ** 함수의 호출 모습을 트리를 사용해 그려보면 아주 간단해짐

마. 재귀적 실행 이해하기(호출과 대기)

아래와 같은 함수를 작성하여 f(5)를 수행한 결과는 무엇인가?

```
void f(int n)
{
    if(n>0) f(n−1);       //n>0 이면 f(n−1) 실행
    printf("%d ", n); //n 값 출력하고 빈칸 1칸 출력
}
```

문제의 의미	- 재귀 함수의 실행에 대한 구체적/명확한 이해. 함수가 호출되어 실행하던 도중 계산이 완료되지 않으면 이후의 명령은 그 계산이 끝날 때까지 실행되지 않음. 재귀함수라면 스택에 저장해둠.
풀이 예시	- n>0이면 이후의 명령수행을 잠시 멈추고, f(n-1) 다시 호출. - f(5)를 실행하면, 5출력을 미뤄두고 f(4) 호출.. 계속 재귀적 반복... f(0)이 호출되면 0 출력하고 미뤄두었던 작업 하나씩 꺼내어 실행. 재귀호출 도중의 계산들은 스택에 저장되어 쌓아짐. - 문제에서 재귀 함수 호출 형태는 2진 사향트리(한쪽으로만 늘어서있는 2진 트리)에서 왼쪽자식은 함수 호출, 오른쪽 자식은 인쇄라고 했을 때, 후위 순회(Left-Right-Root 순으로 방문) 형태를 가지게 됨.

6. 알고리즘 이해하기(코드 이해하기)

가. 정렬 이해하기(선택 정렬)

다음은 어떤 프로그램의 일부이다. n이 6이고, 배열 a가 아래와 같을 때 다음 프로그램을 실행시킨 후 배열 a의 값을 a[1]부터 a[6]까지 차례로 쓴 것으로 알맞은 것은?

a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
2	14	20	8	17	13

```
for(i=1; i<=n-1; i++)        //i는 1부터 n-1까지 1씩 증가
    for(j=i+1; j<=n; j++) //j는 j+1부터 n까지 1씩 증가
        if(a[i] < a[j])    //a[i]<a[j] 이면
        {
            t=a[j]; a[j]=a[i]; a[i]=t;  //두 수의 자리 교환
        }
```

문제의 의미	- 소스코드를 보고 구체적인 데이터의 이동 파악하기
풀이 예시	왼쪽부터 한 자리씩 고정해서 오른쪽의 수들과 차례대로 비교해 나감. 오른쪽에 있는 수들 중에서 왼쪽에 고정해둔 수보다 큰 수가 발견되면, 두 수의 자리를 서로 바꿈. 내림차순 선택정렬의 결과는?

나. 정렬 과정 이해하기(삽입 정렬)

배열 a가 아래와 같고 다음 프로그램으로 처리하려고 할 때, ㉠으로 표시된 줄은 총 몇 번 수행되는가?

a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
10	7	20	5	30	4

```
for(i=2; i<=6; i++)                  //i는 2부터 6까지 1씩 증가
    for(j=i; j>=2; j--)              //j는 i부터 2일 때까지 1씩 감소
        if(a[j] > a[j-1]) break; //오른쪽수가 크면 반복 정지
        else                         //아니면
        {
            t=a[j]; a[j]=a[j-1]; a[j-1]=t;    ...㉠  //교환
        }
```

문제의 의미	- 소스코드를 보고 구체적인 데이터의 이동 파악하기
풀이 예시	- 두 번째 자리의 수부터 시작해서 오른쪽으로 한 칸씩 계산해감. - 왼쪽 수가 작으면 정지. 아니면 서로 위치를 바꾸고 그 이전 수와 같은 방법으로 비교/교환해 나아감. - 삽입정렬로 파악. 자리교환이 일어나는 횟수는?

7. 문제해결 도전하기1(프로그래밍언어로 해결하기)

가. 자료구조 만들기

1에서 50까지의 서로 다른 숫자가 적혀있는 카드 50장이 쌓여있다. 위에서부터 i번째 위치에 어떤 숫자 카드가 있는지를 나타내는 배열이 입력으로 주어졌을 때, 숫자 p가 적힌 카드가 위에서 몇 번째 위치에 있는지를 출력하는 프로그램을 만들려고 한다. ㉠, ㉡에 들어갈 내용을 순서대로 작성한 것은?

```
//a[i]: 위에서부터 I번째에 있는 숫자 카드에 적힌 번호
//p : 찾을 카드 번호
int b[51];
int i;
for(i=1; i<=50; i++) [  ㉠  ];
printf("%d\n", [  ㉡  ]);
```

① b[a[i]]=a[i], a[b[p]]　② b[a[i]]=i, a[b[p]]　③ b[i]=a[i], b[p]
④ b[a[i]]=a[i], b[p]　⑤ b[a[i]]=i, b[p]

문제의 의미	- 입력되어있는 데이터들을 쉽게 다룰 수 있는 자료구조 만들기의 예시. 2중 참조를 이용해 원하는 데이터의 위치로 바로 접근 가능
풀이 예시	- a[] 배열에 임의의 숫자카드가 쌓여있는 순서가 기록되어있음. - b[] 배열에는 어떤 숫자의 카드가 몇 번째에 쌓여있는지 넣으려고 함 - 상황을 더 쉽게 이해하기 위해 예시 그림 그려보면 - a[] : (0: -, 1: 3, 2: 2, 3: 4, 4: 5, 5: 1) - b[] : (0: -, 1: 5, 2: 2, 3: 1, 4: 3, 5: 4) - b[] 배열에 위와 같이 값을 채우기 위해서는 a[] 배열을 순서대로 뒤지면서 b[a[i]]=i 하면 됨. b[]배열을 다 채우면 p값 카드의 위치는 b[p]로 참조 가능.

a[] :

0	1	2	3	4	5
-	3	2	4	5	1

b[] :

0	1	2	3	4	5
-	5	2	1	3	4

나. 핵심 식 만들기

배열 a에 10개의 정수가 들어있다. 이 배열은 오름차순으로 정렬되어 있으며, 몇 개의 중복된 값을 가질 수도 있다. 이 때, 배열 a가 몇 개의 서로 다른 값으로 구성되어 있는 지를 구하는 프로그램을 작성하였다.

예를 들어 배열 a에 다음과 같은 값들이 저장되어있다고 하면 3, 5, 6, 8, 10, 11, 12, 13의 8가지의 값으로 이루어져 있으므로 8이 출력된다. 빈 칸에 들어갈 내용으로 알맞은 것은?

a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]
3	5	5	6	8	10	11	12	13	13

```
p=1;
for(i=2; i<=10; i++) if(          ) p=p+1;
printf("%d\n", p);
```

① a[i]==a[i-1]　② a[i]>a[i-1]　③ a[i]>=a[i-1]
④ a[i]<a[i-1]　⑤ a[i]<=a[i-1]

문제의 의미	- 입력되어있는 데이터들의 형태를 이용해 원하는 것을 얻어내기 위한 아이디어 만들어보기
풀이 예시	- 배열에 오름차순으로 들어있으므로 수의 종류를 알아내기 위해서는 어떻게? - 배열을 순서대로 살펴보는데 왼쪽 배열에 들어있는 수와 다르면 1씩 증가. - 오름차순으로 들어있으므로 다른 것을 판단하기 위해서 a[i-1]<a[i] 검사

다. 반복구조로 해결하기

영어 단어 "madam"이나 "radar"처럼 앞에서 뒤로 읽는 것이나 뒤에서 앞으로 읽는 것이 동일한 단어를 회문(palindrome)이라고 한다. 주어진 문자열이 회문인지를 판별하는 프로그램을 다음과 같이 작성하였다. 빈 칸에 들어갈 내용으로 알맞은 것은?

```
//s :주어진 문자열, n : 그 문자열의 길이
palin=1;
for(i=0; i<=n/2-1; i++)
    if(          ) palin=0;
if(palin==1) printf("회문이다.\n");
else printf("회문이 아니다.\n");
```

① s[i]!=s[n-i-1]　② s[i]!=s[n-i]　③ s[i]!=s[i+n/2]
④ s[n/2-i]!=s[n/2+i]　⑤ s[n/2-i-1]!=s[n/2+i]

문제의 의미	- 입력되어있는 데이터들의 형태를 확인하기 위해 반복구조를 효과적으로 사용하기
풀이 예시	- 앞으로 읽으나, 뒤로 읽으나 같은 단어나 문장을 회문(팰린드롬)이라고 함. - 만약 회문이라면 맨 앞과 뒤가 같은지 한 문자씩 안쪽으로 가면서 확인해 보면 됨. - 계산은 반까지만 하면 모두 확인 가능함. - 가장 앞과 뒤를 순서대로 비교하기 위해 i를 하나씩 증가시키면서 반까지만 비교해보면 됨. s[i]!=s[n-1-i] 검사필요. ** 반복으로 반까지만 검사하기 위해서는 (마지막-처음+1)/2까지만 계산하면 최적. ((n-1)-(0)+1)/2. 왜? 1~10의 정수 개수는? 1~5의 정수 개수는?

라. 재귀함수로 해결하기

영어 단어 “madam”이나 “radar”처럼 앞에서 뒤로 읽는 것이나 뒤에서 앞으로 읽는 것이 동일한 단어를 회문(palindrome)이라고 한다. 주어진 문자열이 회문인지 판별하는 재귀함수 f를 다음과 같이 정의하였다. 재귀함수 f는 주어진 문자열이 회문인 경우에는 1을 돌려주고, 그렇지 않은 경우에는 0을 돌려준다. 이 함수를 이용해 “madam”이 회문인지 검사하기 위해서는 f(“madam”, 0, 4);와 같이 호출하는데, 첫 번째 것은 검사하고자하는 문자열이며, 두 번째는 정수 0이고, 세 번째는 문자열의 길이에서 1을 뺀 값이다. ㉠, ㉡에 들어갈 내용을 순서대로 작성한 것은?

```
int f(char s[], int start, int end)
{
    if( [   ㉠   ] ) return 1;
    else if( [   ㉡   ] ) return 0;
    else return f(s, start+1, end-1);
}
```

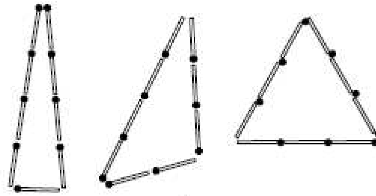
① s[start]==s[end], s[start]!=s[end]　② s[start]==s[end], end<=start+1
③ s[start]==s[end], end<=start　④ end<=start+1, s[start]!=s[end]
⑤ end<=start, s[start]!=s[end]

문제의 의미	- 이전과 이후의 관계를 이용한 재귀함수 설계 연습
풀이 예시	- 만약 왼쪽 문자와 오른쪽 문자가 다르다면 회문이 아님. - 왼쪽 한 문자의 위치를 start, 오른쪽 한 문자의 위치를 end라고 하고 - 같으면 한 칸씩 가운데로 이동시킴. - 재귀적 반복을 중지하기 위해서 end<=start가 되는 경우 중지.

마. 조합적 반복으로 해결하기1(계산식으로 복잡도 줄이기)

같은 길이의 성냥개비가 여러 개 주어져 있다. 이것들을 평면에 늘어놓아서 삼각형을 만들려고 한다. 삼각형의 한 변은 여러 개의 성냥개비를 직선으로 이어서 만들 수 있지만, 성냥개비를 꺾거나 잘라서 변의 한 부분을 만들 수는 없다. 성냥개비의 개수 n이 주어졌을 때, 이들 성냥개비를 사용해 만들 수 있는 서로 다른 삼각형의 개수를 출력하는 프로그램을 작성하시오.

예를 들어 9개의 성냥개비로 만들 수 있는 서로 다른 삼각형은 그림1과 같이 3가지이다.

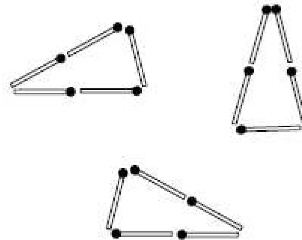

그림 1

주의사항

- 주어진 성냥개비는 모두 사용하여 하나의 삼각형을 만들어야 한다.

- 삼각형을 한 개도 만들 수 없으면 0을 출력한다. 예를 들어서, 주어진 성냥개비의 개수가 1, 2, 3 또는 4인 경우에는 삼각형을 한 개도 만들 수 없다.

- 합동인 삼각형들은 같은 삼각형으로 본다. 예를 들어서 성냥개비 5개를 사용해 만들 수 있는 그림 2의 삼각형들은 같은 삼각형으로 본다.


그림 2

```
//프로그램 일부
c=0;
for(i=1; i<=n; i++)                    //가장 작은 길이 선택
    for(j=[   ㉠   ]; j<=n; j++)  //두 번째 길이 선택
    {
        k=n-i-j;                         //나머지 길이 계산
        if([   ㉡   ] && i+j>k) c++; //삼각형 가능 확인
    }

printf("%d\n", c);
```

위의 빈 칸 ㉠에 알맞은 내용은?

① 1　　② 2　　③ i-1　　④ i　　⑤ i+1

위의 빈 칸 ㉡에 알맞은 내용은?

① i<k　　② i<=k　　③ j<k　　④ j<=k　　⑤ j>k

문제의 의미	- 서로 다른 경우를 찾아내기 위해 크기순으로 배열된 조합을 순서대로 만들어 검사해 보는 연습. 3중 반복구조를 사용하지 않고 2중 반복 구조를 이용해 시간복잡도를 효과적으로 줄이는 예시. - 아이디어를 이해하는 것이 중요
풀이 예시	- 중복하여 계산하지 않도록 크기가 커지도록 선택하는 조합으로 문제 해결 - 가장 작은 한 길이를 먼저 선택 - 두 번째 길이는 첫 번째 길이와 같거나 큼 - 세 번째 길이는 총길이에서 앞의 두 길이의 합을 빼면 됨. - 단, 세 번째 길이가 앞의 두 길이의 합보다 크다면 삼각형이 되지 않음.

바. 조합적 반복으로 해결하기2(반대로 선택하기)

1부터 10사이의 숫자가 적혀진 다섯 장의 카드가 a[]배열에 주어진다. 그 중 3장의 카드를 골라 그 합의 일의 자리수를 가장 크게 만드려고 한다.

예를 들어 다섯 장의 카드가 (7, 5, 5, 4, 9)인 경우 (7, 4, 9)를 선택하면 합이 20이 되어 일의 자리수는 0이 되고, (5, 5, 9)를 선택하면 합이 19가 되어 일의 자리수는 9가 된다. 세 장의 카드를 뽑아 그 합의 일의 자리수를 9보다 크게 만들 수는 없으므로 구하고자 하는 답은 9가 된다.

이와 같이 구한 일의 자리수의 최댓값 m을 구하는 프로그램을 작성하시오.

```
//프로그램 일부
s=0;
for(i=1; i<=5; i++)      //5장 카드 숫자의 총합 계산
    s+=a[i];

m=0;                     //최댓값 m을 0으로 저장
for(i=1; i<=4; i++)      //제외할 첫 번째 카드 선택
    for(j=i+1; j<=5; j++) //제외할 두 번째 카드 선택
    {
        p=(          )%10; //남은 카드 3개합의 일의 자리
        if(p>m) m=p;
    }

printf("%d\n", m);
```

위의 빈 칸에 알맞은 내용은?

① s+a[i]+a[j] ② s+a[i]-a[j] ③ s-a[i]+a[j]

④ s-a[i]-a[j] ⑤ s-a[j-i]

문제의 의미	- 서로 다른 경우를 찾아내기 위해 크기순으로 배열된 조합을 순서대로 만들어 검사해 보는 연습. 3개를 골라내는 3중 반복 구조가 아니라, 2개를 고르는 2중 반복 구조로 시간복잡도를 효과적으로 줄이는 예시.
풀이 예시	- 카드를 순서대로 2개 고르는 아이디어로 계산 - 먼저 모든 카드의 합을 계산 - 그 중 순서대로 2개를 고르고, 합에서 제외하면 3개를 순서대로 선택한 것과 같은 효과가 남 - 생각을 전환하여 반복구조 1개를 효과적으로 줄임

8. 문제해결 도전하기2(알고리즘설계로 빠르게 해결하기)

가. 재귀함수로 분기하기

여행을 떠난 영수는 지도를 하나 구하였다. 이 지도는 아래 그림과 같이 직사각형 모양이며 여러 칸으로 나뉘어져 있다. 한 칸은 한 지점을 나타내는데 각 칸에는 그 지점의 높이가 쓰여 있으며, 각 지점 사이의 이동은 지도상에서 상하좌우로 이웃한 곳끼리만 가능하다.

50	45	37	32	30
35	50	40	20	25
30	30	25	17	28
27	24	22	15	10

현재 제일 왼쪽 이 칸이 나타내는 지점에 있는 영수는 제일 오른쪽 아래 칸이 나타내는 지점으로 가려고 한다. 그런데 가능한 힘을 적게 들이고 싶어 항상 높이가 더 낮은 지점으로만 이동하여 목표 지점까지 가고자 한다. 위와 같은 지도에서는 다음과 같은 3가지 경로가 가능하다.

50	45	37	32	30
35	50	40	20	25
30	30	25	17	28
27	24	22	15	10

50	45	37	32	30
35	50	40	20	25
30	30	25	17	28
27	24	22	15	10

50	45	37	32	30
35	50	40	20	25
30	30	25	17	28
27	24	22	15	10

세로의 크기 m, 가로의 크기 n인 지도가 주어질 때 이와 같이 제일 왼쪽 위 지점에서 출발하여 제일 오른쪽 아래 지점까지 항상 내리막길로만 이동하는 경로의 개수를 구하는 프로그램을 작성하시오.

```
//프로그램 일부
//h[i][j] :지도에서 i행 j열에 위치한 칸이 나타내는 지점 높이
...
int f(int y, int x)
{
    if(s[y][x]>0) return s[y][x];
    else
    {
        int t=0;
        if(y>=2 && h[y-1][x]<h[y][x]) t+=f(y-1,x);
        if(y<=m-1 && h[y+1][x]<h[y][x]) t+=f(y+1,x);
        if(x>=2 && h[y][x-1]<h[y][x]) t+=f(y,x-1);
        if(x>=n && h[y][x+1]<h[y][x]) t+=f(y,x+1);
        s[y][x]=t;
        return t;
    }
}
...
for(i=0; i<=m; i++)
    for(j=0; j<=n; j++)
        s[i][j]=0;
[          ];
printf("%d\n", f(1,1);
```

위 빈 칸에 알맞은 내용은?

① s[1][1]=1 ② s[1][1]=2 ③ s[1][1]=h[1][1]

④ s[m][n]=1 ⑤ s[m][n]=h[m][n]

문제의 의미	- 경로탐색을 위해 재귀함수의 분기를 활용. 재귀함수 내부에서 각각의 재귀적 호출은 각 단계에서 선택할 수 있는 서로 다른 경로로 진행하는 것을 의미함. 재귀함수의 반복적인 호출로 데이터 테이블에 내용을 채워 문제 상황을 해결. 데이터 테이블과 재귀함수의 호출관계를 이해하는 것이 목표. 테이블 완성의 결과가 문제에서 얻고자하는 최종 값이 되도록 테이블 내용 작성 알고리즘 구성. 핵심은 재귀함수의 설계. 주의할 점은 경계 조건의 설정.
풀이 예시	- h[][] 배열의 데이터를 이용해 비교(2차원 높이 데이터)하고 s[][] 배열에 값을 기록하는 4원 분기 재귀함수 f(y, x). 만약 s[y][x]에 기록되어있는 값이 0보다 크면 분기를 중지하고 그 값을 리턴. - 처음에 s[][] 배열을 모두 0으로 채움. 빈칸 실행. - f(1,1)을 호출한 결과를 값으로 출력. - 재귀함수의 분기 순서와 의미 : 높이 데이터를 아래, 위, 왼쪽, 오른쪽의 순서로 차례대로 비교하고, 그쪽이 낮으면 재귀함수 호출함. 그 위치에서의 s[y][x] 값은 이동하는 곳에서 계산된 결과를 누적함.

- h[][] :

0	0	0	0	0	0
0	50	45	37	32	30
0	35	50	40	20	25
0	30	30	25	17	28
0	27	24	22	15	10

를 재귀함수의 호출에 따라 s[][] 채우기

- s[][] 의 처음 상태 :

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

재귀함수를 따라 이동해보면..

0	0	0	0	0	0
0	???	←	←	←	←
0	↑	0	0	↑	↑
0	↑	0	0	↑	0
0	↑	←	←	←	?

3가지 경로 찾아냄

오른쪽 마지막에서 돌아오게 하려면 그 위치의 값이 0보다 커야 함. 만약 그 값이 1이라면?

0	0	0	0	0	0
0	**3**	2	2	2	1
0	1	0	0	1	1
0	1	0	0	1	0
0	1	1	1	1	1

으로 채워지고 리턴되는 값은 3, 경로의 개수!!

** GIS(지리정보시스템, 등고선) 프로그램에서 고도 데이터가 주어지는 경우 위 방법으로 모든 하산 경로를 찾아낼 수 있음!!

나. 배열로 계산 누적하기

어떤 투자가가 여러 기업들에게 돈을 투자해서 최대의 이익을 얻고자 한다. 단, 투자는 만원 단위로 할 수 있으며 각 기업은 많이 투자할수록 많은 이익을 투자가에게 돌려준다. 돈을 투자하지 않은 경우는 당연히 얻게 되는 이익도 없다. 예를 들어서, 한 투자가가 4만원을 갖고 두 개의 기업들에 각각 만원 단위로 투자했을 경우 얻을 수 있는 이익은 다음과 같다.

투자액수 (만원)	기업 A	기업 B
0	0	0
1	5	1
2	6	5
3	7	9
4	10	15

위의 경우 만일 기업 A에 1만원, 기업 B에 3만원을 투자하는 경우 투자가가 얻는

이익은 14만원(5만원+9만원)이다. 4만원을 투자해서 가장 많은 이익을 얻을 수 있는 경우는 B기업에만 4만원을 투자하는 경우로서 이때의 이익은 15만원이다. 여기서 투자가는 한 기업에 돈을 나누어서 투자할 수는 없다.

투자액이 m만원이고, 기업의 개수가 n개, i만원을 j번째 기업에 투자해서 얻는 이익을 p[i][j]만원이라고 할 때 얻을 수 있는 가장 많은 이익금을 출력하는 프로그램을 작성하시오.

```
//프로그램 일부
for(i=0; i<=m; i++)
    for(j=0; j<=n; j++) s[i][j]=0;
for(i=1; i<=n; i++)
    for(j=1; j<=m; j++)
    {
        max=0;
        for(k=[    ㉠    ]; k<=[    ㉡    ]; k++)
        {
            c=[    ㉢    ];
            if(c>max) max=c;
        }
        s[j][i]=max;
    }
printf("%d\n", s[m][n]);
```

위 빈 칸 ㉠, ㉡에 각각 알맞은 내용은?

① 0, j-1 ② 0, j ③ 1, j-1 ④ 1, j ⑤ 1, j+1

위 빈 칸 ㉢에 알맞은 내용은?

① p[j][i]+s[j-k][i-1] ② p[j][i]+s[k-1][i-1] ③ p[k][i]+s[j-k][i-1]

④ p[k][i]+s[k-1][i-1] ⑤ p[j][i]+s[j-1][i-1]

문제의 의미	- 주어진 데이터를 이용해 2차원 배열로 저장될 수 있는 표를 만들고, 그 표를 이용해 또 다른 2차원 배열의 값을 능동적(계산)으로 채워나가는 동적계획법(다이나믹 프로그래밍)의 예시. 표를 만들어 이전 단계에서 만들어진 최적값을 이용해 다음 단계의 또 다른 최적값을 순서대로 만들어 채워나가는 표계산이 핵심. 표를 구성하고 표의 내용을 채우기 위한 알고리즘을 구성하는 것이 핵심. 이러한 동적계획법은 일반적인 인간의 DNA 표준 구조를 해석해내는 어려운 문제를 예상 기간보다 훨씬 빠르게 해결해낼 수 있는 핵심이 되었음.(인간 게놈 프로젝트. 바이오인포매틱스 영역에서 정보과학의 가능성과 힘을 세상에 알린 중요 사건) 검색 : 인간게놈프로젝트 중 dynamic programming 예시 참고(2개의 인간 DNA 염기서열의 공통점 추출)

풀이 예시	각 기업에 따른 투자 대 이익 표현을 위한 p[][] 배열 투자 대 최대 이익 계산 결과를 계산해서 누적하기 위한 s[][] 배열 처음에 s[][] 배열 모두 0으로 초기화 중첩 반복문으로 1만원 투자했을 때의 최댓값을 s[][]배열의 가장 오른쪽에 나오도록 구성

p[][] :

	0	1	2
0	0	0	0
1	0	5	1
2	0	6	5
3	0	7	9
4	0	10	15

를 이용해 s[][] 채우기

s[][] 의 처음 상태 :

	0	1	2
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0

처음 n만원을 첫 번째 기업에 투자한 최대 이익 결과

	0	1	2
0	0	0	0
1	0	5	0
2	0	6	0
3	0	7	0
4	0	10	0

,

처음 시작해서 아래쪽으로 가면서 p[][]을 참조하고 가능한 최대 이익을 계산해 아래쪽으로 기록해 감. 각 셀에는 n만원을 1번 회사에 투자했을 때의 최대 이익이 나오게 됨.

두 번째 열, 회사가 2개 일 때 최대 이익 s[2][2]를 얻으려면?

p[][]:

	0	1	2
0	0	0	0
1	0	5	1
2	0	6	5
3	0	7	9
4	0	10	15

, s[][]:

	0	1	2
0	0	0	0
1	0	5	5
2	0	6	**?**
3	0	7	-
4	0	10	-

. 같은 색끼리 더한 것 중 최대를 ?에 기록하면 된다. 생각해보면 2만원을 조합할 수 있는 경우인 (0,2), (1,1), (2,0) 조합이 됨. ?위의 5는 (0,1), (1,0) 중 큰 값을 먼저 기록해 둔 상태임

전체적으로 살펴보면, 이전 회사들만 가지고 구했던 최댓값들과 추가되는 회사의 값들로 조합이 가능한 경우에 대한 이익들 중에서 가장 큰 값을 찾아 기록해가는 형태

이해 연습	

STEP1 : p[][]:

	0	1	2
0	0	0	0
1	0	5	1
2	0	6	5
3	0	7	9
4	0	10	15

, s[][]:

	0	1	2
0	0	0	0
1	0	5	5
2	0	6	**?**
3	0	7	-
4	0	10	-

STEP2 : p[][]:

	0	1	2
0	0	0	0
1	0	5	1
2	0	6	5
3	0	7	9
4	0	10	15

, s[][]:

	0	1	2
0	0	0	0
1	0	5	5
2	0	6	
3	0	7	
4	0	10	

STEP3 : p[][]:

	0	1	2
0	0	0	0
1	0	5	1
2	0	6	5
3	0	7	9
4	0	10	15

, s[][]:

	0	1	2
0	0	0	0
1	0	5	5
2	0	6	
3	0	7	
4	0	10	

. 결과 최대 이익은?

Ⅲ 정보올림피아드 지필 문제의 해설(2013, 2014, 2015)

2013 시도예선 초등부 문제

1. 빨간 구슬, 노란 구슬, 파란 구슬이 각각 한 개씩 있다. 세 구슬의 무게의 합이 70Kg이고, 빨간 구슬과 노란 구슬의 무게의 합이 40Kg이며, 노란 구슬과 파란 구슬의 무게의 합이 50Kg일 때, 노란 구슬의 무게는 몇 Kg인가?

① 5　② 10　③ 15　④ 20　⑤ 30

** 출제의도?
문제 상황에 대한 추상화. 관계식 만들기. 논리적 해결
** 의미?
복잡하게 보이는 문제 상황을 기호 등을 사용해 추상화하고, 주어진 상황을 표현할 수 있는 관계식을 만들고, 그렇게 만들어진 관계식들을 이용해 문제를 해결하는 논리적인 과정
** 해결예시?
빨간 구슬, 노란 구슬, 파란 구슬을 각각 R, Y, B 라고 추상화하고, 관계식을 만들면
R+Y+B=70,　R+Y=40,　Y+B=50
관계식들을 잘~ 관찰해서 Y의 값을 얻어내면 된다.
(연립방정식이라는 말도 쓸 수 있지만, 본질은 관찰을 통한 문제 해결이다)

2. 1×1+2×2+3×3+…120×120 의 마지막 자리의 숫자(1의 자릿수)는 얼마인가?

① 0　② 1　③ 2　④ 3　⑤ 5

** 출제의도?
반복적 패턴으로 만들어지는 결과에 대한 계산 패턴 분석
** 의미?
모두 계산할 수도 있지만, 아이디어를 이용하면 빠르게 답을 얻어낼 수 있다.
얻어내야 하는 목표를 처음에 명확히 확인하고 시작하는 것이 필요하다.
** 해결예시?
계산 결과의 1의 자리를 맞추는 것이 목표이므로 계산 과정에서 10의 자리 이상은 고려할 필요가 없다.
계산이 복잡할 것 같지만 분명 패턴이 있을 것이라 가정하고, 그 패턴을 찾을 때까지 몇 개 해 본다.
1*1 = 1
2*2 = 4
3*3 = 9
4*4 = 6　(10의 자리 이상은 필요 없음. 목표는 1의 자리 수이다.)
5*5 = 5

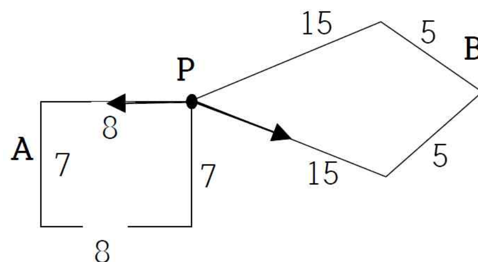
6*6 = 6
7*7 = 9
8*8 = 4
9*9 = 1
10*10 = 0
11*11 = 1 (이제 보니 패턴이 만들어진다.)
찾아낸 패턴은 1 4 9 6 5 6 9 4 1 0 ... 10개의 반복적 패턴
120까지 계속 한다면 위의 패턴이 12번 만들어진다. 이를 이용해서 계산할 때도 쌍으로 묶어 계산하면 빠르다

3. 무게가 서로 다른 구슬 세 개가 있다. 양팔저울을 사용하여 가장 가벼운 구슬, 중간 무게의 구슬, 가장 무거운 구슬의 순서대로 나열하고 싶다. 양팔저울을 최소 몇 번 사용하면 항상 나열할 수 있는가?

① 2 ② 3 ③ 4 ④ 5 ⑤ 6

** 출제의도?
어떤 문제의 답을 반드시 얻기 위해 필요한 최소 계산 횟수에 대한 이해
** 의미?
경험적으로도 해결할 수 있지만, 논리적이면서 체계적인 사고 방법이 필요하다. 예를 들어 구슬이 4개라면 어떻게 할 것인가? 이는 가장 간단한 정렬 방법으로도 유도할 수 있다.
** 해결예시?
A, B, C 라고 하고, "가장 가벼운 것을 먼저 찾고, 나머지에서 다시 가장 가벼운 것을 찾아나가는 방법을 사용하면.. " 반드시 찾을 수 있다.
따라서, A:B, A:C 를 통해서 가장 가벼운 것 하나 찾고. 2개가 남으면 1번만 더 달면 된다.

4. 철수와 영희는 점 P에서 동시에 출발하여 달리기를 한다. 철수는 경로 A를 따라 달리고, 영희는 경로 B를 따라 철수와 같은 빠르기로 달린다. 변 옆의 수는 변의 길이를 나타낸다. 철수가 10바퀴 돌았을 때 영희와 몇 번 만났는가? 단, 처음 출발할 때 만난 것은 제외한다.



① 0 ② 1 ③ 2 ④ 3 ⑤ 4

** 출제의도?

상황분석, 패턴 분석

** 의미?

반복적인 패턴이 있을 때, 특별한 상황 찾기?

** 해결예시?

P에서 시작해서 한 바퀴 돌아올 때까지 각각 달리는 상황을 살펴보면

철수 A : 8 7 8 7 --> 합 30

영희 B : 15 5 5 15 --> 합 40

분석한 상황을 해석하면, 철수가 한 바퀴 돌 시간에, 영희는 10만큼 모자란다.

그렇다면 철수가 4바퀴 돌았을 때 영희를 만나게 된다.

5. 30명의 학생들에게 두 문제가 주어졌다. 1번 문제를 푼 학생이 25명이고, 2번 문제를 푼 학생이 20명이다. 이 경우, 두 문제 모두 푼 학생 수의 최소와 최대는 각각 얼마인가?

① 15, 30 ② 10, 15 ③ 5, 15 ④ 5, 20 ⑤ 15, 20

** 출제의도?

논리, 집합, 상황 파악, 가정?

** 의미?

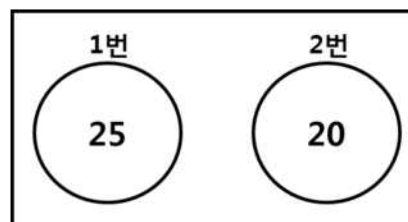
최대 최소 문제는 프로그래밍 언어를 활용하는 문제 해결에서 가장 일반적인 문제이기도 하다. "최적화 문제"

이러한 문제들의 경우 여러 가지 상황에 대한 최선/최악의 경우들을 정확하게 파악하는 것이 필요하다.

** 해결예시?

이런 문제의 경우 다이어그램을 그리는 것이 이해를 아주 쉽게 할 수 있다.

‘이런 경우가 있을까?’ ‘총인원이 30인데?’ ‘다른 경우는?’과 같은 질문을 해 본다.



6. 상자에 빨간색 왼쪽 장갑 5개, 빨간색 오른쪽 장갑 5개, 노란색 왼쪽 장갑 3개, 노란색 오른쪽 장갑 3개, 흰색 왼쪽 장갑 2개, 흰색 오른쪽 장갑 2개가 들어있다. 이제 상자에 손을 넣어 장갑을 하나씩 꺼내려 한다. 단, 한번 꺼낸 장갑은 다시 상자에 넣지 않는다. 상자에서 장갑을 꺼낸 후에만 장갑의 색을 알 수 있다고 한다. 그러면 적어도 몇 개의 장갑을 꺼내야만 같은 색의 왼쪽, 오른쪽 장갑 한 쌍을 항상 꺼낼 수 있을까?

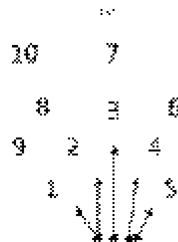
① 5 ② 9 ③ 11 ④ 15 ⑤ 19

** 출제의도?
추상화, 상황 단순화, 최악의 경우 찾기?
** 의미?
어떤 상황에서 최악의 경우를 생각하는 의미. 최악의 경우는 어떤 알고리즘으로 문제를 해결할 때 걸릴 수 있는 최대 시간을 가늠할 수 있게 한다. 문제 해결을 위해 최악의 경우를 가정하는 것은 어떤 형태의 알고리즘으로 해결해야 하는가와 직접적인 연관성을 가진다.
** 해결예시?
빨간 장갑 A, 노란 장갑 B, 흰색 장갑 C 라고 하고 각각 왼손 l, 오른손 r 이라고 하면.
문제 상황은
Al : 5, Bl : 3, Cl : 2
Ar : 5, Br : 3, Cr : 2
최악의 경우는 어느 한쪽 손 장갑만 모두 꺼내는 경우이다. 그 다음에 꺼낼 때에는 무조건 쌍이 맞는다.

7. 왼손의 손가락은 순서대로 엄지, 검지, 중지, 인지, 약지라고 부른다. 철수가 왼손을이용하여 1부터 100까지 100개의 수를 차례대로 다음과 같이 센다. 먼저 엄지부터 시작하여 수를 세는 데, 엄지는 1, 검지는 2, 중지는 3, 인지는 4, 약지는 5라고 센다. 그다음 방향을 바꿔서 인지 6, 중지 7, 검지 8, 엄지 9라고 세고, 다시 방향을 바꿔 검지 10, 중지 11, ... 이런 식으로 센다. 그러면 100은 어떤 손가락일까?

① 엄지 ② 검지 ③ 중지 ④ 인지 ⑤ 약지

** 출제의도?
수의 크기가 크기 않기 때문에, 그냥 해봐도 적당히 되긴 한다.
하지만, 패턴 분석과 거꾸로 사고가 필요하다.
** 의미?
상황을 바라보는 관점을 역으로 바꾸는 방법으로 쉽게 해결 가능하다.
** 해결예시?
패턴을 찾기 위해 몇 개 해 보고, 그 상황을 분석해 패턴을 찾아낸다. 패턴이 안 나오면 거꾸로 생각해 본다.



상황을 그대로 따라가면 그림과 같은 형식이 된다. 이를 손가락의 입장에서 살펴보자.
손가락을 a, b, c, d, e 라고 하면 만들어지는 패턴은 a, b, c, d, e, d, c, b, a, b,
손가락 입장에서 반복되는 패턴은 a, b, c, d, e, d, c, b ... 의 8개가 반복적으로 나타난다. 그렇다면 100까지는 8개 주기로 12개 들어가고, 나머지 4개가 남는다.
그렇다면 a, b, c, d .. 즉, d 손가락이 선택될 것이다.

8. 길이가 100cm인 줄자가 하나 있다. 이 줄자를 길이가 1cm인 100개의 조각들로 나누고 싶다. 여러 개의 줄자 조각들을 모아 한 번에 자를 수 있다고 할 때, 최소 몇 번만 자르면 가능할까? 단, 줄자는 접을 수 없다.

① 7 ② 8 ③ 9 ④ 10 ⑤ 99

** 출제의도?

반씩 줄여가는 분할 정복, 즉 로그의 개념적 의미를 이해하는가?

** 의미?

2개의 그룹이나 갯수로 구분하는 경우가 문제 해결에서는 매우 많다. 그랬을 때 나누어지지 않는 부분은 어떻게 할 것인가?

최악의 경우까지 생각해 봐야하므로 나누어진 것에 대해서, 큰 것을 기준으로 잡는 것이 합리적이다.

** 해결예시?

100을 2개로 나누면 50, 50을 2개로 나누면 25, 25를 2개로 나누면 12, 13 이 되어야 하는데(1cm 로 나누어야 하므로) 그 중 큰 것으로 해야 최악의 경우까지 생각 가능하다.

13을 2개, 7을 2개..

이런 식으로 계속가면 자르는 최소 횟수를 찾을 수 있다.

9. 1부터 100까지 모든 정수의 각 자리에 나타난 수를 모두 더하면 얼마인가?

① 899 ② 900 ③ 901 ④ 902 ⑤ 903

** 출제의도?

경우의 수에 대한 논리적 사고, 패턴 찾기?

** 의미?

물론, 모두 더한 다음에 각자리 숫자를 더해도 된다. 하지만, 매우 비효율적인 방법이다.

어떤 패턴을 찾는 다면 계산을 아주 빠르고 효과적으로 할 수 있다.

** 해결예시?

1부터 100까지 어떤 숫자 ?가 나타나는 패턴은 단지 3가지 뿐 이다. (# : 0~9)

한 자리 숫자 ?
두 자리 숫자 ?#
두 자리 숫자 #?
세 자리 숫자 ?##

이 모든 경우를 모두 찾아 더하면 그 수가 들어가는 모든 경우에서 그 수를 더한 것과 같게 된다.

그럼 1은? 한 자리 숫자로는 1개, 두 자리 숫자로는 10개+9개(0?는 아니다.), 세 자리 숫자로는 1개.. 총 21개

그럼 2는? 1개, 19개, 0개.. 20개

그럼 3은? 1개, 19개, 0개.. 20개

그럼 4는?

다음과 같이 빠르게 계산할 수 있다 : (1+2+3+...+7+8+9)*20 + 1

10. 1부터 시작하여, 이전의 수에 2를 곱하거나, 이전의 수에 1을 더하는 연산을 이용하여 숫자들을 차례로 만든다고 하자. 다음은 13을 만드는 두 방법을 보여준다.

방법1: 1 → 2 → 3 → 6 → 12 → 13
방법2: 1 → 2 → 4 → 5 → 6 → 12 → 13

주어진 두 가지 연산만을 이용하여 13을 만드는 방법은 위의 방법 이외에도 여러 가지있다. 그 중에서, 13을 만드는 최소 연산 횟수는 위의 방법1과 같이 5이다. 250을 만드는데 필요한 최소 연산 횟수는 얼마인가?

① 8 ② 9 ③ 10 ④ 11 ⑤ 12

** 출제의도?
상태 트리? 재귀함수?, 오토마타? 최적해(최소) 찾기?
** 의미?
어떤 상태에서 다른 상태로 변화할 수 있는 여러 가지 경우가 있을 경우, 상태 트리/솔루션 트리를 그릴 수 있다.
이는 재귀함수를 이용한 문제해결에서 기본적으로 사용되는 방법이다.
하지만, n개의 가능한 상태가 있을 때, 2^n 개씩 늘어나므로 매우 커진다.
또한, 어떤 상태는 다른 상태에 의해서도 만들어질 수 있다.
그래서 본질은 솔루션 그래프.. 상태 그래프로 최적화 되는 것들이다.
그럼 어떻게 할 것인가?
** 해결예시?
첫 번째 전략은.. 1부터 250까지 만드는 모든 경우를 찾아내는 것이 아니라..... *2, +1
거꾸로 250에서 1로 만들어가는 방향으로 가는 것이다. .... /2, -1
그러면 생각해야할 것이 매우 많이 줄어든다.
그런데 막상 이렇게 그려보면 생각했던 것보다 빠르게 찾을 수가 없다.
여기에서 논리적인 생각을 적용한다.
'나누는 상황이 있고, 하나를 빼는 상황이 있는데, 하나를 빼는 상황이 반으로 나눈 상황보다 더 빠르게 250에 접근할까?'
'만약 아래로 계속 내려간다면 언젠가는 1이 되긴 할까?(적어도 무조건 1개씩 빼는 상황이니 무조건 된다.)'
이 때, 다음과 같은 전략이 가능하다.
짝수면 2로 나눈다. 그리고 또 나눈다..
홀수가 되면 1을 뺀다. 그리고 또 같은 방법으로 나누어 내려간다.
언제까지? 1이 될 때까지
이 것이 최소이다. (증명이 필요하지만, -1보다 /2가 더 빠르게 접근해 감을 볼 때 가능함을 알 수 있다.)
이렇게 생각하면.. 아래처럼 가능하다.
250
125 124
62
31 30
15 14

7 6

3 2

1

계산 횟수를 세어보면 된다. 거꾸로 접근, 상태 트리. 전략적인 접근으로 매우 쉽게 해결 가능한 문제이다.

11. 아래 표에 나타난 100개의 수를 모두 더하면 얼마인가?

1	2	3	4	5	6	7	8	9	10
2	3	4	5	6	7	8	9	10	11
3	4	5	6	7	8	9	10	11	12
4	5	6	7	8	9	10	11	12	13
5	6	7	8	9	10	11	12	13	14
6	7	8	9	10	11	12	13	14	15
7	8	9	10	11	12	13	14	15	16
8	9	10	11	12	13	14	15	16	17
9	10	11	12	13	14	15	16	17	18
10	11	12	13	14	15	16	17	18	19

① 550 ② 850 ③ 950 ④ 1000 ⑤ 1050

** 출제의도?

패턴 찾기/이용, 부분 합 아이디어

** 의미?

프로그래밍 언어를 활용해 문제를 효율적으로 해결하기 위해서는 보다 적은 계산에 대한 생각과 아이디어가 필요하다.

물론 위의 표를 다 더할 수는 있다, 그런데. 단순 계산하는 것은 계산의 횟수가 너무 많다. 패턴을 보고, 변화되는 내용들을 보고, 그 상황을 이용하면 아주 쉽게 계산이 가능하다.

** 해결예시?

1	2	3	4	5	6	7	8	9	10
2	3	4	5	6	7	8	9	10	11
3	4	5	6	7	8	9	10	11	12
4	5	6	7	8	9	10	11	12	13
5	6	7	8	9	10	11	12	13	14
6	7	8	9	10	11	12	13	14	15
7	8	9	10	11	12	13	14	15	16
8	9	10	11	12	13	14	15	16	17
9	10	11	12	13	14	15	16	17	18
10	11	12	13	14	15	16	17	18	19

합 55

첫 번째 줄의 합은 55이다. 그런데 오른쪽 줄을 보면 1이 빠지고 11이 들어왔다.

그럼 오른쪽 줄은 왼쪽 줄 합에 10만큼 더 커진 것이다.

그럼 그 오른쪽은 두 번째 줄에 10만큼 더 커진 것이다.

결국 계산은 55(처음 한 줄의 합)*10줄 + 변화량의 합(10+20+30+...+90)이 된다.

12. A, B, C, D, E 다섯 개의 문자로만 이루어진 메시지를 주고받으려 한다. 메시지를 주고받기 전에 A는 00, B는 01, C는 100, D는 101, E는 11로 나타내었다. 예를 들어, 메시지 BBD를 보내려면 0101101을 보내면 된다. 1001010110110000이라는 메시지를 받았다면, 이 메시지에 있는 문자는 모두 몇 개인가?

① 4 ② 5 ③ 6 ④ 7 ⑤ 8

** 출제의도?

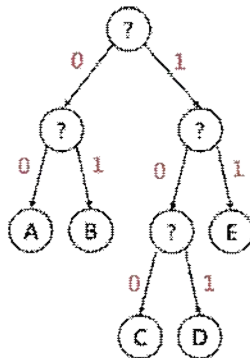
허프만 코딩의 이해?

** 의미?

모르스 부호나, 허프만 코딩은 어떤 기호에 고정 폭의 암호화비트를 할당하는 것이 아니라, 많이 사용하는 기호에 보다 적은 비트를 할당해 전체적인 입장으로 보았을 때, 전송량을 줄이기 위한 아이디어이다. 문제는 허프만 코딩에 대한 내용이라고 할 수 있다.

문제를 약간 바꿔서, 비트가 서로 겹치도록 할 수도 있긴 하겠다. 가능한 문자는? 가능한 경우의 개수는?

위 상황을 허프만 트리로 그려보면 아래와 같다.



트리의 루트에서 링크를 따라가면서 거기에 적혀있는 비트를 읽는다.

잘 보면, 문자들은 트리의 리프(터미널)로 배치된다. 그렇기 때문에 서로 코드가 겹치지 않는 것이다.

** 해결예시?

허프만 트리라는 것이 확정되면 그냥 앞에서부터 읽어 가면 된다.

13. 네 명의 학생 갑, 을, 병, 정이 100m 달리기를 하였다. 친구들이 달리기 결과를 다음과 같이 예측하였다.

(친구1) 1등:갑, 2등:을, 3등:병
(친구2) 1등:을, 2등:병, 3등:정
(친구3) 1등:병, 2등:갑, 3등:정

경기가 끝났을 때, 위의 친구 세 명은 각각 한 명의 등수만 정확하게 맞추었다. 또한 세 명의 친구가 맞춘 등수는 모두 달랐다. 이 때, 2등한 학생은 누구인가?

① 갑 ② 을 ③ 병 ④ 정 ⑤ 알 수 없음

** 출제의도?
체계적인 논리 검증?
** 의미?
프로그래밍을 통해서 논리적인 해결 과정들을 거치게 되는데, 그런데 필요한 체계적인 논리적 사고와 가정/검증에 대한 문제이다. 주어진 상황을 읽어, 확정된 사실을 찾아내고 그것을 이용해 논리적으로 검증하면 된다.
** 해결예시?
예를 들어,
'친구1이 정확하다' 라고 했을 때. 1등이 정확했다면?
2등이 정확했다면?
3등이 정확했다면?
이렇게 가정하고 확인하고 체계적으로 따라가 보면 된다.

14. 다음과 같이 숫자가 표시된 카드 11장이 있다. 이 카드들 중에서 여러 장을 뽑아서 뽑힌 카드들의 숫자를 모두 더하여 하나의 수를 만들려고 한다. 만들 수 없는 수들 중에서 가장 작은 수는?

1	1	2	5	7	17	17	35	83	170	340

① 89 ② 129 ③ 169 ④ 339 ⑤ 449

** 출제의도?
경우의 수? 조합? 하지만, 그렇게 보지 말도록 하자.
** 의미?
여러 가지 경우들 중에서 어떤 조건을 만족하는 것을 찾아내야 할 때가 있다. 그 상황에서 또 최선인지를 확인해야 한다.
아이디어들이 필요한데, 물론 주어진 예를 만들 수 있는 것들을 시행착오적으로 계산해보면서 생각할 수도 있지만, 문제 상황이 더 크고 복잡하다면 한번 선 처리를 하고 생각하면 어떨까?
** 해결예시?
몇 개만 선택하는 것보다 일단, 어느 부분까지 모두 다 선택했다고 고려하자.
그리고, 그렇게 선택하고 왼쪽에서 오른쪽으로 가면서 차례대로 모두 선택한다고 하자.
예를 들어 카드를 5개 집는다고 가정하면 일단 왼쪽 4개를 모두 잡고 나머지 중에서 한 개만 선택하는 방법을 취할 수 있는 것이다.

카드	1	1	2	5	7	17	17	35	83	170	340
합	1	2	4	9	16	33	50	85	168	338	678

합은 왼쪽 카드부터 그 카드 까지 모두 선택해 합했을 때의 수이다.
이렇게 해두면 확인하기가 매우 편리하다? 그냥 카드를 몇 개 선택해서 어떤 수를 만들 수 있나 확인해보고, 아니면, 왼쪽부터 모든 카드를 다 들었을 때를 가정하고 몇 개를 빼보는 방법으로도 확인할 수 있다.
예를 들어 89를 만들 수 있는지 확인하려면 83에 6을 만들면 된다.
129는 83까지 모든 카드를 다 들면 168이 되는데, 39만큼 차이가 나므로 38까지의 모든 카드들을 들었을 때 39만큼 카드를 빼낼 수 있으면 만들 수 있다.

15. 세 야구팀 갑, 을, 병이 서로 한 번씩의 경기를 하였다. 갑은 두 경기에서 4득점을 하였고, 4실점을 하였다. 을은 두 경기에서 4득점을 하였고, 3실점을 하였다. 병은 두 경기에서 4득점을 하였고, 5실점을 하였다. 갑과 병 사이의 경기 점수로 가능한 것은?

① 3:2 ② 2:2 ③ 4:2 ④ 2:1 ⑤ 3:1

** 출제의도?
상황 이해, 추상화, 표현식 만들기?
** 의미?
복잡한 것 같은 문제 상황을 보다 쉽게 이해할 수 있도록 효과적으로 표현하고, 그 추상화된 내용들을 통해 문제 해결
** 해결예시?
갑을병 팀을 각각 A, B, C 라고 하면, 가능한 게임은?(왼쪽부터 한개를 고정 시키고, 나머지를 선택해 가는 방법을 쓰면 체계적으로 모든 경우를 찾아낼 수 있다. 4팀이라면?
A : B
A : C
B : C
이런 상황에서 게임 스코어를 생각해 보자. 게임 스코어를 다시 적당한 문자로 바꾸어 넣으면,
a : b
c : d
e : f
자 그럼 주어진 문제에서 갑팀, 즉, A 팀이 두 번 싸워서 4득점, 4실점 이라고 했다면
a+c = 4, b+c=4 의 식이 하나 만들어진다.
...
병팀 즉 C 팀에 대해서는 4득점, 5실점이니
d+f = 4, c+e = 5
...
계산해보면.. a+b+c+d+e+f=12 적당히 빼고 더하면 각 스코어를 구할 수 있다.
(미지수가 6개이고, 서로 다른 식이 6개이면 무조건 미지수의 값을 구할 수 있다.)

16. 다음 프로그램의 출력 결과는 무엇인가?

```
int y=2013, m=5, d=25;
printf("%d\n", y+m*d);
```

① 2013 ② 2018 ③ 2038 ④ 2043 ⑤ 2138

** 출제의도?
연산자의 계산 순서
** 의미?

소스 코드의 간단한 이해

** 해결예시?

2013 + 5 * 25

17. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, tot=0;
for(i=2010; i<=2013; i++) tot=tot+3*(i-2000)+1;
printf("%d\n", tot);
```

① 140 ② 142 ③ 144 ④ 146 ⑤ 148

** 출제의도?

단순, 부분 코드의 의미 이해

** 의미?

코드의 부분 부분들이 어떤 일들을 하는지 이해하고 적용할 수 있는지에 대한 문제

** 해결예시?

i 는 2010 부터 2013까지

tot = tot + ... 는 이전 결과에 더한다는 의미로 해석

3*(i-2000) + 1 는 2000빼고, 3곱하고, 더하기 1 한 것.

그렇다면.

2010 2011 2012 2013

31 34 37 40 ... <- 의 합을 구하는 문제이다.

18. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, y=0;
for(i=2000; i<=2013; i++){
  if(i%400==0) y++;
  else if(i%4==0 && i%100!=0) y++;
}
printf("%d\n", y);
```

① 4 ② 5 ③ 6 ④ 7 ⑤ 8

** 출제의도?

조건 실행 구조와 논리식의 이해

** 의미?

윤년 판단식을 조건 실행 구조와 논리식으로 구성한 것

** 해결예시?

2000 ~ 2013 까지로 몇 개 안되니 400으로 나누어지면 윤년, 4의 배수이면서 100으로 나누어 떨어지지 않으면 윤년이다.

참고 : 왜 이렇게 윤년을 계산하는가? 지구 공전 주기는 정확히 365일이 아니고

365.2564 일이다.
따라서 1년을 365일로 잡으면, 0.2564... 일씩 모자라게 된다.
그렇기 때문에.. 대략 4년 마다 하루를 늘려주면 약 1.0256 일이 되기 때문에 적당히 보상된다.

19. 다음 프로그램의 출력 결과는 무엇인가?

```
int i=1, count=0;
while(i*i<=150){
  count++;
  i++;
}
printf("%d\n", count);
```

① 10 ② 11 ③ 12 ④ 13 ⑤ 14

** 출제의도?
코드의 이해
** 의미?
어떤 수의 제곱근(루트)을 구하는 조금 다른 방법의 일부?
루트(150) 보다 작은 정수의 갯수를 구하는 코드이다.
루트(150)을 구해서 작은 것을 찾는 것이나. 어떤 수를 제곱해서 150보다 작은 것을 보는 것이나 같다.
** 해결예시?
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 까지는 제곱하면 150보다 작다.

20. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, sum=0;
for(i=1; i<=2013; i+=13){
  if(i%2==0) sum=sum+1;
  else sum=sum+2;
}
printf("%d\n", sum);
```

① 155 ② 156 ③ 232 ④ 233 ⑤ 310

** 출제의도?
코드의 이해, 패턴분석, 임계조건(조건의 양 끝단을 주의해야 한다.)
** 의미?
반복적으로 실행되는 코드에 대한 이해, 패턴 분석, 결과 예측
** 해결예시?
1부터 2013 까지 13씩 증가함.. 1부터 2013까지 13씩 더해지는 각각에 대해서, 홀수이

면 +1, 짝수이면 +2, 패턴을 살펴보기 위해 몇 개 적어본다

1 14 27 40 53 … … 13(k-1)+1 형태로 간다.

홀 짝 홀..

0 13 26 39 52 … …

13으로 나눠지는 모듈이 154개 만들어지고, 거기에 11개 더 한 모듈이 홀짝으로 더해지니 3만큼… 이를 이용해서 계산하면 된다.

21. 다음 프로그램의 출력 결과는 무엇인가?

```
int sum=0, a=2013;
while(a>0){
  sum+=a%10;
  a/=10;
}
printf("%d\n", sum);
```

① 1 ② 2 ③ 3 ④ 5 ⑤ 6

** 출제의도?

코드의 이해

** 의미?

변수 변화를 따라가면 안 되고, 항상 "이게 무엇을 수행하는 코드인가?"라는 관점에서 바라보아야 한다. 어떤 수의 마지막 1의 자리 수를 하나씩 잘라내 더하는, 즉 각 자리 수의 합을 구하는 코드이다.

** 해결예시?

2 + 0 + 1 + 3

22. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, a=2, b=3, c=5;
for(i=2; i<65; i++){
  if(i%a==0 && i%b==0 && i%c==0) break;
}
printf("%d\n", i);
```

① 6 ② 15 ③ 30 ④ 60 ⑤ 65

** 출제의도?

코드의 이해. 어떤 기능을 하는 것인가?

프로그래밍은 수학적 특징이나 성질을 확인하기 매우 편리하다.

그래서 수학적 특징이나 의미 성질을 확인하는 코드들이 더러 나온다.

** 의미?

2부터 64까지 한 개씩 올려보면서

2, 3, 5 로 모두 나누어지는 처음 수가 나오면 정지한다.
결국 최소 공배수를 구하게 된다.
** 해결예시?
거꾸로 2*3*5 로 계산하면 된다.

23. f(5)의 값은 무엇인가?

```
int f(int n){
  int i, tot=0;
  for(i=1; i<=n; i++) tot=tot+i;
  return tot;
}
```

① 10 ② 11 ③ 13 ④ 15 ⑤ 21

** 출제의도?
함수의 이해. 이 함수가 어떤 짓을 하는 함수인가?
** 의미?
함수 하나는 어떤 자동화된 작업을 하는 모듈로도 생각할 수 있지만,
문제 해결 프로그래밍에서는... 어떤 문제 한 개에 대한 답을 의미할 수도 있다.
따라서.. f(n) 이라는 함수는 n이라는 상황에서 얻어진 답 한 개라고 생각하는 것이 중요하다.
f(n)은 1부터n까지 정수 합을 계산하는 함수이다.
** 해결예시?
f(5) = 1 + 2 + 3 + 4 + 5

24. f(10)의 값은 무엇인가?

```
int f(int n){
  int i, j, ret=0;
  for(i=1; i<n; i++){
    for(j=1; j<2*i; j++) ret=ret+1;
  }
  return ret;
}
```

① 45 ② 55 ③ 81 ④ 90 ⑤ 100

** 출제의도?
함수의 이해. 이 함수가 어떤 짓을 하는 함수인가?
패턴 분석 및 이해, 목표에 대한 이해?
** 의미?
단순 함수에 대한 이해도 있지만, 함수의 패턴이 어떻게 만들어지는지에 대한 이해가 필

요한 문제이다. 만들어지는 패턴을 이해하기 위해서 몇 개 해 본다.
무엇을 구하는 건지 살펴본다.
그 다음에 만들어지는 패턴에 대해서. 구하는 것을 어떻게 빠르게 구할지 고민한다.

** 해결예시?

몇 개의 패턴을 만들어 보면

```
1   2   3   4   5   6   7   8 ...
1   1   1   1   ..
    2   2   2
    3   3   3
        4   4
        5   5
            6
            7
```

1부터 n까지.. 수가 만들어지는 패턴인데, 그 수의 개수의 총합이니, 1개, 3개, 5개, 7개, 9개..... 17개(9인 경우) 모두 더하면 된다. 더할 때에도 짝지어 더하면 빠르다

25. 다음 프로그램의 출력 결과는 무엇인가?

```
int L[14]={6,-3,2,-7,4,-2,3,10,-2,1,-3,-4,5,-1};
int S[14], M, i;
S[0]=L[0];
for(i=1; i<14; i++){
  if(S[i-1]>0) S[i]=S[i-1]+L[i];
  else S[i]=L[i];
}
M=S[0];
for(i=1; i<14; i++) if(M<S[i]) M=S[i];
printf("%d\n", M);
```

① 14　② 15　③ 16　④ 17　⑤ 18

** 출제의도?

배열 채우기 방법의 이해, 이전의 값을 읽어 동적으로 채워나감.
코드의 이해. 코드가 무슨 기능을 하는가?

** 의미?

문제 해결에서 효과적으로 사용되는 방법인 동적계획, 동적프로그래밍 방법과 관련한 기초적인 코드 이해 문제이다.
표의 특정 상태에 있는 값들을 이용해 표의 데이터를 채워나가는 코드이다.
이전에 만들어진 최적해를 이용해 다음 최적해를 쌓아 올라간다는 기본적인 아이디어와 관련이 있다.

** 해결예시?

어떤 n번째 S[n] 배열의 값이 0보다 크면.. 이전 S배열에 L배열의 값을 더해 다음 S 배열을 채워나간다.

만약 0보다 작다면, 그냥 L 배열의 값을 채운다.

그리고 그 S 배열을 찾아서 최댓값 찾기이다. 거의 완벽한 다이나믹 프로그래밍의 해결 과정이다.

따라가보면

L 배열 : 6 -3 2 -7 4 -2 3 10 -2 1 -3 -4 5 -1

S 배열 : 6 3 5 -2 4 .... 채워나간다. 직접 그려보면 이해가 빠르다.

26. 배열 a에는 다음과 같은 값이 주어진다.

```
int a[]={1,37,3,10,13,23,4,8,17};
```

아래 주어진 함수 SomeAverage(a, 9, 5)를 호출했을 때 출력결과는 무엇인가?

```
void SomeAverage(int a[], int n, int m){
  int i, j, sum;
  for(i=0; i<=n-m; i++){
    sum=0;
    for(j=i; j<i+m; j++) sum=sum+a[j];
      printf("%d ", sum/m);
  }
  printf("\n");
}
```

① 13 17 11 12 13　　② 12 17 10 11 13

③ 12 17 11 12 13　　④ 12 17 10 12 13

⑤ 12.8 17.2 10.6 11.6 13

** 출제의도?

함수의 이해, 의미 분석

** 의미?

무엇을 하는 코드인가? 유심히 살펴보면, 배열을 전달받아서 5개씩 부분합을 만들고, 그 평균(정수/정수 잘림)을 출력하는 함수이다.

패턴을 찾기 위해 처음 몇 개를 직접 손으로 그려보면 된다.

** 해결예시?

```
          1 37 3 10 13  23 4 8 17    <- 9개의 원소
5개합                 64
                        86 <- 모두 더할 필요가 없이 64에서 1빼고, 23 더하면 된다. 변화만 보면 된다.
                          53
                            58
                              65
```

5로 나눈 평균(정수) 이므로 순서대로 12, 17, 10, 11, 13가 된다.

27. f(3, 4)의 값은 무엇인가?

```
int f(int x, int n){
  if(n==1) return 1;
  else return x*f(x, n-1);
}
```

① 1 ② 3 ③ 9 ④ 27 ⑤ 81

** 출제의도?

재귀함수의 이해, 단순재귀(재귀함수 안에 1개만 재귀 호출 되는 경우)

** 의미?

재귀함수가 분할되어 실행되는 상태에 대한 명확한 이해를 필요로 한다.

재귀함수의 분할 형태와 실제 실행에 대해서 이해하기 위해서는 함수에 대한 명확한 이해가 필요하다.

함수는 값을 전달받을 수 있고, 전달 받은 값을 이용해 위에서 아래로 순서대로 실행된다.

실행되는 과정에서 재귀 호출이 일어나면 그 아래로 진행하지 못하고 재귀 호출을 실행한다.

그 재귀 호출이 완료되어야 그 아래 명령들이 실행된다.

여러 가지 재귀 형태가 있을 수 있지만, 재귀함수가 1개만 호출되는 형태이고,

재귀 호출한 결과를 알아야 원래 호출 함수의 결과를 계산할 수 있다.

이는 마치, 어떤 문제 한 개에 대한 답을 얻기 위해 보다 작은 문제로 나누어가는 것과 유사하다. 재귀 함수의 실행 형태를 명확히 이해하기 위해서 일종의 상태 트리를 그려서 재귀 함수의 분할과 실행 과정을 이해할 수 있다. 계산 결과를 얻을 때까지, 현재 상태에서 진행을 멈추고 계속 내려가야 한다. 결과를 얻고 그 값이 return 되면 거꾸로 올라가면서 멈추었던 계산들이 실행되고 최종적으로 답이 나온다.

** 해결예시?

원하는 문제의 결과를 f(3, 4)라고 하면 트리를 그릴 수 있다.

28. f(4)의 값은 무엇인가?

```
int f(int n){
  if(n<=0) return 1;
  else return f(n-1)+f(n-3)*3;
}
```

① 4 ② 7 ③ 10 ④ 14 ⑤ 22

** 출제의도?

재귀 함수의 이해. 다중 재귀(2개 이상의 재귀 호출)

** 의미?
재귀 함수는 일반적인 문제 해결에서 매우 쉽게 프로그래밍 할 수 있다.
하나의 문제를 보다 작은 문제로 나누어 해결하고자 하는 전략이다.
재귀 함수를 설계하기 위해서는 문제를 구성하는 보다 작은 재귀적 관계를 찾고, 그것을 표현하고, 그 관계가 언제 끝나야하는지만? 언제까지 파고 들어서 언제 끝낼지만 작성해 주면 된다.
백트랙킹의 기본.
현재 상황에서 가능한 모든 경우들을 가지치기로 해 볼 수 있는 가장 간단한 방법 등을 찾는다.
재귀적으로 호출되는 관계들 중에서 가장 효율적인 값이나 경우를 선택적으로 골라 계산하든지, 더하든지 빼든지 등의 작업들이 가능하다.
재귀적 호출 관계를 그림으로 그려보면 확실해진다.
** 주의 : 맹목적으로 재귀함수가 비효율적이라는 선입견을 가질 수 있다. 하지만, 그렇기도 하고 아니기도 하다. 일단 그래도 표현이 매우 간단하고, 복잡한 계산 관계는 컴퓨터가 하면 된다. 사람은 그 관계성과 언제까지 계산할지만 작성해 주면 끝난다.
이러한 다중 재귀는 모든 문제 해결에 매우 효과적으로 사용될 수 있다.
** 해결예시?
함수를 트리 형태로 그려 살펴볼 수 있다.

29. f(10, 20)을 실행한 후, count의 값은 무엇인가?

```
int count=0;
int f(int a, int b){
  if(a==b){
    count++;
    return a;
  }
  else{
    int mid;
    mid=(a+b)/2;
    return f(a, mid)+f(mid + 1, b);
  }
}
```

① 11 ② 20 ③ 50 ④ 122 ⑤ 165

** 출제의도?
다중 재귀의 이해, 코드의 이해
** 의미?
다중 재귀는 여러 가지 경우로 분할 재귀 호출 할 수 있는데, 분할된 결과들 중에서 골라내거나, 계산하는 등의 작업이 가능하다. 또한 재귀함수의 비효율성을 줄이기 위해 배열에 값을 기록해 해결하는 메모이제이션 방법을 사용할 수 있다.
분기하는 과정에서 '만약에 그 값이 있다면 더 이상 분기할 필요가 없다'가 아이디어이다.

백트랙킹에 다이나믹 프로그래밍(테이블 채우기) 방법을 사용한다고 해서 다니나믹컬 백트랙킹이라는 말로도 불리는 최신 이론이다.
** 해결예시?
구간을 나누는 과정만 주의하고 28번의 상태 트리(재귀함수의 호출 관계)를 그려보고 계산하면 된다.
f(10, 20)은
f(10, 15) f(16, 20) 의 2개로 나누어지고, 나중에는 그 결과를 합친다.

30. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[15]={0,1,1,0,1,0,0,1,0,1,1,1,0,1,1};
void f(int n){
  if(n>=15) return;
  if(a[n]!=1) return;
  a[n]=2;
  f(n+1);
}
int main(){
  int i, g;
  g=0;
  for(i=0; i<15; i++){
    if(a[i]==1){
      f(i);
      g=g+1;
    }
  }
  printf("%d\n", g);
  return 0;
}
```

① 1　② 3　③ 5　④ 7　⑤ 9

** 출제의도?
재귀 함수의 이해. 문제 해결 프로그래밍
** 의미?
f(n) 이라는 함수의 의미를 먼저 파악해야 한다.
f(n) 은 그 위치 n에 들어있는 배열의 값이 1이면 그 배열에 2를 채우고 다음 배열 값을 보면서 나아가는 코드이다.
배열에 들어있는 값이 1이 아니면 중단 --> 연속된 1을 따라가는 함수이다.
main() 안에 들어있는 루틴을 보면 1이 적힌 부분만 f(n) 함수를 호출하는데, 재귀 함수의 동작을 따라가 보면, 결국 1이 연속으로 붙어있는 그룹의 갯수를 찾는 프로그램이다.
** 해결예시?

처음 상태
0 1 1 0 1 0 0 1 0 1 1 1 0 1 1
재귀함수가 호출되어 한번 돌아가면.
0 2 2 0 1 0 0 1 0 1 1 1 0 1 1
순차적으로 1을 찾다가 한번 더 돌아가면
0 2 2 0 2 0 0 1 0 1 1 1 0 1 1
...
0 2 2 0 2 0 0 2 0 2 2 2 0 2 2
g = g + 1 은 결국 1이 그룹 되어있는 갯수를 카운팅하기 위한 변수로 사용되었다.

31. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[20]={1,4,3,2,4,1,2,5,1,2,3,5,4,1,2,1,1,1,1,9};
int i, s=0, gr=0;
for(i=0; i<20; i++){
  if(a[i]+s>10){
    gr++;
    s=0;
  }
  else s=s+a[i];
}
if(s>0){
  s=0;
  gr++;
}
printf("%d\n", gr);
```

① 1 ② 4 ③ 5 ④ 6 ⑤ 20

** 출제의도?
코드의 이해
** 의미?
별다른 의미는 없는 것으로 보이지만, 계속 더해가다가 일정 크기를 넘으면 카운팅하는 코드이다.
** 해결예시?
처음에 합 s를 0으로 놓고, 더해가기 시작한다. 그러다가 10을 넘으면 카운팅 올리고 합 s를 0으로 리셋한다.

```
1 4 3  2  4 1 2 5 1  2  3 5  4  1 2 1 1 1 1 9
1 5 8 10 14    <-- 카운팅
                1 3 8 9 12  <-- 카운팅
                               3 8 12  <-- 카운팅
                                          1 3 4 5 6 7 16 <-- 카운팅
```

32. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, x=2, y=2, xx, yy;
for(i=0; i<2013; i++){
  xx=-y;
  yy=x;
  x=xx;
  y=yy;
}
printf("%d %d\n", x, y);
```

① 2 2　　② 2 -2　　③ -2 -2　　④ -2 2　　⑤ 0 2

** 출제의도?
  코드의 이해, 패턴 찾기, 패턴 활용하여 예측하기
** 의미?
  패턴이 만들어질 때까지 몇 개를 돌려보면서 패턴을 찾아낸다.
  패턴을 이용해 출력될 값에 대한 예상을 한다.
** 해결예시?

```
i    0    1    2    3      4    5 ...
xx  -2   -2    2    2  |  -2    ...
yy   2   -2   -2    2  |   2    ...
x   -2   -2    2    2  |  -2    ...
y    2   -2   -2    2  |   2    ...
```

4가지 패턴이 반복적으로 나오게 된다. 그럼 2012일 때는 어떤 세트가 될까?

33. 다음 보기 중에서 가장 큰 값은 무엇인가?

```
int next[10]={4,1,9,0,3,2,6,5,8,7};
int f(int n){
  int i, k;
  i=next[n], k=1;
  while(i!=n){
    i=next[i];
    k=k+1;
  }
  return k;
}
```

① f(0)　　② f(1)　　③ f(2)　　④ f(3)　　⑤ f(6)

** 출제의도?

함수의 이해. 인덱싱 따라가기/데이터 찾아가기

** 의미?

주어진 코드의 동작과정을 이해하고 그 방법에 따라 움직여보기.

코드에서는 주어진 값을 찾기 위해 배열의 인덱스를 이용해서 찾는데, 처음에 주어진 값을 찾아내면 그때까지 점프 뛴 횟수를 출력하는 코드이다. 일종의 인덱싱을 통한 값 찾기와 관련된 소스 코드이다.

** 해결예시?

f(0) : next[0]=>4 이므로 next[4]=>3 이므로, next[3]=>0 .. 찾음.. 3번 인덱스 찾는다. 결과는 3이다.

f(1) : next[1]=>1 이므로 결과는 1이다.

34. f(8, 5)의 값은 무엇인가?

```
int f(int n, int r){
  int p[11][11];
  int i, j;
  for(i=0; i<=n; i++){
    for(j=0; j<=n; j++) p[i][j]=0;
  }
  p[0][0]=1;
  for(i=0; i<=n; i++){
    p[i][0]=1;
    for(j=1; j<=i; j++) p[i][j]=p[i-1][j-1]+p[i-1][j];
  }
  return p[n][r];
}
```

① 56　② 57　③ 58　④ 59　⑤ 60

** 출제의도?

동적계획(동적테이블, 다이나믹 프로그래밍)의 이해.

** 의미?

다이나믹 프로그래밍(표에 수 채워나가기)

이전에 만들어져있는 표안의 값들을 이용해 다음 데이터를 동적으로 채워간다.

표에 모든 데이터가 정적으로 채워져 있는 것이 아니라, 이전의 상태에 의해 다음 값이 확정되고, 그렇게 만들어진 값이 그 다음의 값을 만들어내는데 사용된다.

보기의 코드는 직접적으로 파스칼의 삼각형을 만들어내는 코드이다.

2차원 배열이 서로 계산되는 모습을 보면, 2차원 테이블에서 값이 어떻게 만들어져 나가는지 찾아낼 수 있다. 어떤 자리의 값은 윗줄 왼쪽의 값 + 윗줄의 값으로 만들어지는 것을 확인하게 된다.

다이나믹 프로그래밍 방법의 구체적 코딩 예시이다.

다이나믹 프로그래밍은 이전에 만들어져 있는 최적 값으로 이후의 최적값을 쌓아 올라가는 방법이다.

** 해결예시?

```
1  0  0  0  0  0  0  0  0      <-- 처음 시작하는 데이터
1  1  0  0  0  0  0  0  0      <-- 좌상+상 의 값으로 데이터 채워나감.
1  2  1  0  0  0  0  0  0
1  3  3  1  0  0  0  0  0
1  4  6  4  1  0  0  0  0
 .....
```

계속 채워보면 된다.

35. 다음 보기 중에서 가장 큰 값은 무엇인가?

```
int f(int n){
  int a[10]={0}, t=0, i;
  while(n>0){
    a[n%10]=1;
    n=n/10;
  }
  for(i=0; i<10; ++i) t=t+a[i]*i;
  return t;
}
```

① f(711587)　② f(999999)　③ f(614331)
④ f(843211)　⑤ f(914312)

** 출제의도?
함수의 이해. 의미 파악
** 의미?
'함수가 어떤 작업을 하는가?'를 파악하는 것이 중요하다.
함수의 내용을 보면 각 자리의 숫자들을 찾아내 체크하고, 체크한 수를 합하는 함수이다.
결국 들어온 수의 각 자리에 사용된 숫자들만 골라내서 그 합을 만들어내는 함수이다.
** 해결예시?
f(999999) 의 결과는 9
f(711587) 의 결과는 1 5 7 8 을 모두 더하면 21이 된다.

36. 다음 보기 중에서 SomeString(str1, str2)를 호출했을 때 0을 반환하는 것은 무엇인가? (단, str1, str2는 char[] 자료형이며, 각 보기는 str1, str2를 차례로 나타낸다.)

```
int SomeString(char a[], char b[]){
  int i, j, n, m;
  n=strlen(a);
```

```
  m=strlen(b);
  if(n!=m) return 0;
  for(i=0; i<n; i++){
    for(j=0; j<m; j++){
      if(a[i]==b[j]){
        b[j]='*';
        break;
      }
    }
  }
  for(i=0; i<m; i++) if(b[i]!='*') return 0;
  return 1;
}
```

① "pizza" "pazzi" ② "listen" "silent"
③ "competition" "emoticontip" ④ "mississippi" "imisspsipis"
⑤ "algorithm" "logtrimat"

** 출제의도?
배열 처리 함수. 함수의 이해
** 의미?
단어가 들어있는 2개의 배열을 받아 처리한다. 단어의 개수가 같아야 하고, 단어의 개수가 같은 경우에는 한 문자씩 비교해서 * 모양으로 체크한다.
마지막에 * 이 아닌 것이 남아있는 경우... 0을 리턴한다.
결국, 주어진 두 단어를 비교해서 각 단어에 쓰여 있는 문자의 개수가 같은가 체크하는 함수이다
** 해결예시?
algorithm
logtrimat --> ********t t가 하나 남는다.

37. SomeNumbers()를 호출했을 때 출력 결과는 무엇인가?

```
void SomeNumbers(){
  int i, j, tot;
  for(i=10; i<100; i++){
    tot=0;
    for(j=1; j<i; j++) if(i%j==0) tot=tot+j;
    if(tot==i) printf("%d\n", i);
  }
}
```

① 10 ② 16 ③ 24 ④ 28 ⑤ 36

** 출제의도?
함수의 의미. 함수의 동작 파악
** 의미?
어떤 수가 들어오면, 그 수보다 1개 작은 수까지 모두 나누어 보고, 나누어지면 나누어지는 수를 더한 다음에 마지막에 그 합을 출력한다.
...
가끔 수학적인 특성을 가진 값 등을 프로그래밍한 예를 제시하기도 하는데,
이 코드는 수학적 관점에서의 완전수(Perfect Number)를 찾아내기 위한 것의 일부이다.
완전수는 자기 자신을 제외한 양의 약수를 더했을 때, 자기 자신이 나오는 양의 정수를 의미한다.
** 해결예시?
10 ? : 1, 2, 5 합은?
16 ? : 1, 2, 4, 8 합은?

38. 다음 프로그램의 출력 결과는 무엇인가?

```
int c=0, i, j, k;
char str[]="akakakka";
int len=8;
int p;
for(i=0; i<len; i++){
  for(j=i; j<len; j++){
    p=1;
    for(k=0; i+k<j-k; k++) {
    if(str[i+k]!=str[j-k]) {
      p=0;
      break;
    }
  }
  if(p==1) c=c+1;
  }
}
printf("%d\n", c);
```

① 13 ② 14 ③ 15 ④ 16 ⑤ 17

** 출제의도?
소스 코드의 이해
** 의미?
1차원 배열을 활용하는 소스코드를 이해해야하는데, 앞으로 읽으나 뒤로 읽으나 같은 회문(팰린드롬)을 찾아내는 소스코드로 확인할 수 있다.
i 는 회문의 가장 왼쪽에 있는 문자의 위치라고 가정하고, 그 오른쪽으로 하나씩 가면서

만들 수 있는 가장 긴 회문의 길이를 찾는다.

마지막까지 다~ 하고 나면, 주어진 문자열에서 잘라내 만들 수 있는 최대 회문의 길이를 골라 낼 수 있게 되는데, 문제에서는 만들 수 있는 회문의 총 갯수를 출력하는 형태이다.

** 해결예시?

a k a k a k k a 가 주어졌을 때, 각 문자를 가장 왼쪽에 두고 만들 수 있는 회문의 갯수를 찾아 기록해 보면

3 a, aka, akaka 가능
3 k, kak, kakak 가능
2 a, aka 가능
2 k, kak 가능
2 a, akka 가능
2 k, kk 가능
1 k 가능
1 a 가능

* 실기 대회 연습 문제(코드 해석만 첨가)

[39-40] 다음과 같은 문제를 해결하기 위해 프로그램을 작성하였다. 물음에 답하시오.

0부터 9까지의 숫자가 표시된 카드를 가지고 두 사람 A와 B가 게임을 한다. A와 B에게는 각각 0에서 9까지의 숫자가 하나씩 표시된 10장의 카드뭉치가 주어진다. 두 사람은 카드를 임의의 순서로 섞은 후 숫자가 보이지 않게 일렬로 늘어놓고 게임을 시작한다. 단, 게임 도중 카드의 순서를 바꿀 수는 없다.

A와 B 각각이 늘어놓은 카드를 뒤집어서 표시된 숫자를 확인하는 것을 한 라운드라고 한다. 게임은 첫 번째 놓인 카드부터 시작하여 순서대로 10번의 라운드로 진행된다. 각 라운드에서는 공개된 숫자가 더 큰 사람이 승자가 된다. 승자에게는 승점 3점이 주어지고 패자에게는 승점이 주어지지 않는다. 만약 공개된 두 숫자가 같아서 비기게 되면, A, B 모두에게 승점 1점이 주어진다.

10번의 라운드가 모두 진행된 후, 총 승점이 큰 사람이 게임의 승자가 된다. 만약, A와 B의 총 승점이 같은 경우에는, 제일 마지막에 이긴 사람을 게임의 승자로 정한다. 그래도 승부가 나지 않는 경우는 모든 라운드에서 비기는 경우뿐이고 이 경우에 두 사람은 비겼다고 한다.

예를 들어, 다음 표에서 3번째 줄은 각 라운드의 승자를 표시하고 있다. 표에서 D는 무승부를 나타낸다. 이 경우에 A의 총 승점은 16점이고, B는 13점이어서, A가 게임의 승자가 된다.

라운드	1	2	3	4	5	6	7	8	9	10
A	4	5	6	7	0	1	2	3	9	8
B	1	2	3	4	5	6	7	8	9	0
승	A	A	A	A	B	B	B	B	D	A

아래 표의 경우에는 A와 B의 총 승점은 13점으로 같다. 마지막으로 승부가 난 라운드는 7번째 라운드이고, 이 라운드의 승자인 B가 게임의 승자가 된다.

라운드	1	2	3	4	5	6	7	8	9	10
A	9	1	7	2	6	3	0	4	8	5
B	6	3	9	2	1	0	7	4	8	5
승	A	B	B	D	A	A	B	D	D	D

A와 B가 늘어놓은 카드의 숫자가 순서대로 주어질 때, 게임의 승자가 A인지 B인지, 또는 비겼는지 결정하는 프로그램을 작성하시오.

입력 형식

입력 파일의 이름은 INPUT.TXT로 한다. 입력 파일은 두 개의 줄로 이루어진다. 첫 번째 줄에는 A가 늘어놓은 카드의 숫자들이 빈칸을 사이에 두고 순서대로 주어진다. 두 번째 줄에는 B가 늘어놓은 카드의 숫자들이 빈칸을 사이에 두고 순서대로 주어진다.

출력 형식

출력 파일의 이름은 OUTPUT.TXT로 한다. 첫 번째 줄에는 게임이 끝난 후, A와 B가 받은 총 승점을 순서대로 빈칸을 사이에 두고 출력한다. 두 번째 줄에는 이긴 사람이 A인지 B인지 결정해서, 이긴 사람을 문자 A 또는 B로 출력한다. 만약 비기는 경우에는 문자 D를 출력한다.

입력과 출력의 예

입력(INPUT.TXT)

```
7 1 6 2 3 0 5 9 4 8
7 1 6 2 3 0 5 9 4 8
```

출력(OUTPUT.TXT)

```
10 10
D
```

프로그램

```
#include <stdio.h>
int A[10], B[10];
//스코어 입력을 위한 전역 배열 선언. 전역은 자동으로 0 초기화
//전역 변수 선언. 지역 배열 보다 크게 잡을 수 있음.

int main(){
  int i, j;

//자신이 필요하다고 생각하는 변수 설정
  int last_winner, A_score, B_score;

//파일 입출력을 위한 파일 포인터.. 루틴..
//파일 읽기 모드로 열기
  FILE *fin, *fout;
  fin=fopen("INPUT.TXT", "rt");

//파일에서 첫번째 줄 10개 읽어들이기
  for(i=0; i<10; i++) fscanf(fin, "%d", &A[i]);

//다음 10개...줄을 구분하지는 않고 읽었음.
  for(i=0; i<10; i++) fscanf(fin, "%d", &B[i]);
```

```
//파일 읽기 종료
  fclose(fin);

//초기 설정
  last_winner=-1;
  A_score=0;
  B_score=0;

//승점 계산
  for(i=0; i<10; i++){
    if(A[i]>B[i]) A_score+=3, last_winner=0;
    else if(A[i]<B[i]) B_score+=3, last_winner=1;
    else A_score++, B_score++;
  }

//파일 출력 준비
  fout=fopen("OUTPUT.TXT", "wt");

//승점 파일로 출력
  fprintf(fout, "%d %d\n", A_score, B_score);

//승점이 같으면?... 논리.. 만들기..
  if(A_score==B_score){
    if( (a) ) fprintf(fout, "D");
    else if ( (b) ) fprintf(fout, "A");
    else fprintf(fout, "B");
  }
  else if(A_score>B_score) fprintf(fout, "A");
  else fprintf(fout, "B");
  fclose(fout);
  return 0;
}
```

39. 다음 중 (a)와 (b)에 들어갈 내용으로 알맞은 것은?

① (a) last_winner == 1
(b) last_winner == 0

② (a) last_winner == −1
(b) last_winner == 0

③ (a) last_winner == −1
(b) last_winner == 1

④ (a) last_winner == 1
(b) last_winner == −1

⑤ (a) last_winner == 0
(b) last_winner == 1

40. 입력 파일이 다음과 같이 주어졌을 때, 위 프로그램의 출력은 무엇인가?

```
9 1 7 2 6 3 0 4 8 5
6 3 9 2 1 0 7 4 8 5
```

① 13 13
B

② 12 14
B

③ 13 11
A

④ 14 12
A

⑤ 13 13
A

2013 시도예선 중고등부 문제

1. 1!+2!+3!+...+10! 의 마지막 자리의 숫자(1의 자릿수)는 얼마인가? 여기서 n! = 1*2*3*4*5 이다. 즉, n!은 1부터 n까지 모든 자연수의 곱이다. 예를 들어, 5!=1*2*3*4*5=120

① 1 ② 3 ③ 5 ④ 7 ⑤ 9

** 출제의도?
목표 / 분석 / 다이나믹 프로그래밍
** 의미?
그냥 보면 복잡할 것 같지만, 1의 자리 숫자만 찾아내면 된다.
** 해결예시?
1! + 2! + 3! + 4! .... ==> 각 항에 대해서 1의 자리 숫자만 보면 1, 2, 6, 4, 나머지는 계속 0이다. 그럼 합하면?

2. 1부터 100까지 모든 정수의 각 자리에 나타난 수를 모두 더하면 얼마인가?

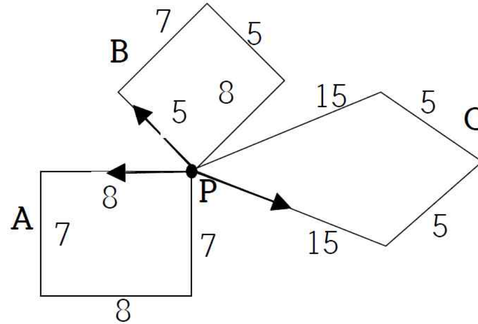
① 899 ② 900 ③ 901 ④ 902 ⑤ 903

** 출제의도?
경우의 수에 대한 논리적 사고, 패턴 찾기?
** 의미?
물론, 모두 더한 다음에 각자리 숫자를 더해도 된다. 하지만, 매우 비효율적인 방법이다.
어떤 패턴을 찾는 다면 계산을 아주 빠르고 효과적으로 할 수 있다.
** 해결예시?
1부터 100까지 어떤 숫자 ?가 나타나는 패턴은 단지 3가지 뿐이다. (# : 0~9)

한 자리 숫자	?
두 자리 숫자	?#
두 자리 숫자	#?
세 자리 숫자	?##

이 모든 경우를 모두 찾아 더하면 그 수가 들어가는 모든 경우에서 그 수를 더한 것과 같게 된다.
그럼 1은? 한 자리 숫자로는 1개, 두 자리 숫자로는 10개+9개(0?는 아니다.), 세자리 숫자로는 1개.. 총 21개
그럼 2는? 1개, 19개, 0개.. 20개
그럼 3은? 1개, 19개, 0개.. 20개
그럼 4는?
빠른 계산 : (1+2+3+...+7+8+9)*20 + 1

3. 철수, 영희, 길동이가 점 P에서 동시에 출발하여 철수는 경로 A를 따라, 영희는 경로 B를 따라, 길동은 경로 C를 따라 달린다. 세 사람이 모두 같은 빠르기로 달린다. 변 옆의 수는 변의 길이를 나타낸다. 철수가 110바퀴 돌았을 때 세 사람이 동시에 몇 번 만났는가? 단, 처음 출발할 때 만난 것은 제외한다.



① 4　② 5　③ 6　④ 7　⑤ 8

** 출제의도?
상황이해, 패턴 분석?
** 의미?
최소공배수 활용?
** 해결예시?
철수 영희, 길동이가 움직이는 경로를 쓰고 관찰한다.
A : 8 7 8 7 -> 30
B : 5 7 5 8 -> 25
C : 15 5 5 15 -> 40
가운데 P 점에서 만나려면 최소 공배수인 600 만큼 움직였을 때 처음 만난다. 철수가 110 바퀴 돌았다는 것은 총 3300 만큼 움직인 것이다.

4. 1000개의 동전이 있다. 이 중 999개의 동전의 무게는 각각 10그램이고, 나머지 동전 하나는 10그램이 아니다. 여러 동전의 무게의 합을 한 번에 정확히 잴 수 있는 저울을 이용하여 10그램이 아닌 동전 하나가 무엇인지 알아내려고 한다. 이를 위해 필요한 저울의 최소 사용 회수는 얼마인가?

① 9　② 10　③ 11　④ 49　⑤ 99

** 출제의도?
로그 개념?
** 의미?
분할 정복?
** 해결예시?
1000, 500, 250, 125, 62, 31, 15, 7, 3, 2 개씩 나눠 보면 최악의 경우에도 무조건 찾을 수 있다.

5. 어떤 함수 f(x)는 모든 정수 x에 대하여 f(x)=f(x-1)+f(x+1)을 만족한다. 만약 f(1)=1 이고, f(2)=-2 이면, f(2013)은 얼마인가?

① -3 ② -2 ③ 1 ④ 2 ⑤ 3

** 출제의도?

관계의 이해, 재귀 함수의 이해, 패턴 분석

** 의미?

** 해결예시?

바닥 f(1), f(2)에서 시작해 몇 개 올려가 본다.

n	1	2	3	4	5	6	7	8	9	10	11 ...	
	1	-2	-3	-1	2	3	1	-2	-3	-1	2	<-----

반복적 패턴이 나타난다. 이를 이용해서 2013번째를 구할 수 있다.

6. 30명의 학생들에게 세 문제가 주어졌다. 1번 문제를 푼 학생이 25명이고, 2번 문제를 푼 학생이 23명이며, 3번 문제를 푼 학생이 18명이면, 세 문제 모두 푼 학생은 최소 몇 명인가?

① 0 ② 5 ③ 6 ④ 7 ⑤ 12

** 출제의도?

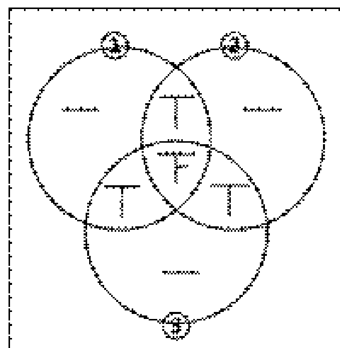
집합과 논리?

** 의미?

최선의 경우, 최악의 경우 찾기?

** 해결예시?

집합과 논리 관련한 문제는 벤다이어그램에 표시해보면 쉽게 구할 수 있는 경우가 많다. 단, 여기서는 최소 최대까지 고려해야한다. 그림을 그리고 바를 정자로 1번 문제 푼사람 영역 체크, 2번 문제 푼사람 영역 체크, 3번 문제 푼사람 영역 체크한다.



저 가운데에 속하는 최소 인원을 계산해야한다. 세문제 모두 푼 학생을 최소화하기 위해 문제를 2개 푼 학생들의 가능한 최소 인원을 계산한다.

1번 2번 푼 학생의 최소는 18명. 왜냐하면 1번 25명+2번 23명 인데. 최대 30명이니 최소 인원은 18명으로 계산한다. 마찬가지로, 2번 3번 푼 학생의 최소는 11명, 1번 3번 푼 학생의 최소는 13명. 그러고 나면.. 저 가운데 최소 인원을 계산할 수 있게 된다.

1번푼사람 + 2번푼사람 + 3번푼사람 - 1/2번최소 - 2/3번최소 - 1/3번최소 하면 가운데가 비는데, 25 + 23 + 18 - 18 - 11 - 13 ==> 24 남는데, 그럼 결국 가운데의 최소는 6명이 된다.

7. 네 명의 학생 갑, 을, 병, 정이 100m 달리기를 하였다. 친구들이 달리기 결과를 다음과 같이 예측하였다.

(친구1) 1등:갑, 2등:을, 3등:병
(친구2) 1등:을, 2등:병, 3등:정
(친구3) 1등:병, 2등:갑, 3등:정

경기가 끝났을 때, 위의 친구 세 명은 각각 한 명의 등수만 정확하게 맞추었다. 또한 세 명의 친구가 맞춘 등수는 모두 달랐다. 이 때, 2등한 학생은 누구인가?

① 갑　② 을　③ 병　④ 정　⑤ 알 수 없음

** 출제의도?

체계적인 논리 검증?

** 의미?

프로그래밍을 통해서 논리적인 해결 과정들을 거치게 되는데, 그런데 필요한 체계적인 논리적 사고와 가정/검증에 대한 문제이다. 주어진 상황을 읽어, 확정된 사실을 찾아내고 그것을 이용해 논리적으로 검증하면 된다.

** 해결예시?

예를 들어,

친구1이 정확하다.. 라고 했을 때. 1등이 정확했다면?

2등이 정확했다면?

3등이 정확했다면?

이렇게 가정하고 확인하고 체계적으로 따라가 보면 된다.

8. 1부터 999까지 모든 자연수들을 사전적 순서로 나열할 때, 350번째의 수는 무엇인가? 예를 들어, 10, 11, 21, 100, 101, 111, 120을 사전적 순서로 나열하면 10, 100, 101, 11, 111, 120, 21이 된다.

① 41　② 350　③ 405　④ 413　⑤ 514

** 출제의도?

정열개념, 경우의 수, 논리적 사고

** 의미?

꼼꼼하게 체계적으로 경우의 수를 따져 채워나가면 편함

** 해결예시?

몇 가지 있나? 오름차순으로 처음에 몇 개 해본다 1~999

1 : 1개

1 # : 10 개

1 # #   : 100 개 --> 한 숫자에 대해서 111개 있음. 그럼 350번째는? 3까지 돌리면 333개 이고, 나머지를 해보면,

4 : 1개

4 0 : 1개

4 0 # : 10개 ...

41

410

411

412

413 <-- 몇 번째일까?

9. 무게가 서로 다른 구슬 세 개가 있다. 양팔저울을 사용하여 가장 가벼운 구슬, 중간 무게의 구슬, 가장 무거운 구슬의 순서대로 나열하고 싶다. 양팔저울을 최소 몇 번 사용하면 항상 나열할 수 있는가?

① 2 ② 3 ③ 4 ④ 5 ⑤ 6

** 출제의도?

어떤 문제의 답을 반드시 얻기 위해 필요한 최소 계산 횟수에 대한 이해

** 의미?

경험적으로도 해결할 수 있지만, 논리적이면서 체계적인 사고 방법이 필요. 예를 들어 구슬이 4개라면? 어떻게 할 것인가? 이는 가장 간단한 정열 방법으로도 유도할 수 있다.

** 해결예시?

A, B, C 라고 하고, "가장 가벼운 것을 먼저 찾고, 나머지에서 다시 가장 가벼운 것을 찾아나가는 방법을 사용하면.. " 반드시 찾을 수 있다.

따라서, A:B, A:C 를 통해서 가장 가벼운 것 하나 찾고. 2개가 남으면 1번만 더 달면 된다.

10. 열자리 십진수 '7abcdefg77'은 1900과 2000 사이의 어떤 정수의 세제곱이다. g에 해당하는 숫자는 무엇인가?

① 1 ② 2 ③ 4 ④ 8 ⑤ 9

** 출제의도?

분석, 가능한 경우 줄여 나가기...

** 의미?

가려진 문자나 기호의 수를 맞추는 복면산과 비슷하지만, 가능한 경우들이 아주 많을 수 있고, 또 손 계산이 매부 복잡하다. 주어진 형태나 패턴을 보고 거꾸로 가능한 경우의 수를 줄여 나가야 한다.

** 해결예시?

첫자리가 7이므로 2000은 아님을 알 수 있다.

가장 뒷자리가 7이니 어떤 수가 가능한지 샘플링 해 본다. 단, 세제곱을 하더라도 1의 자리만 확인하면 되기 때문에

1 2 3 4 5 6 7 8 9 0

제곱 : 1 4 9 6 5 6 9 4 1 0 ....
세제곱 : 1 8 7 4 5 6 3 2 9 0 ....   --> 세제곱을 해서 끝자리가 7이 되려면 반드시 3인 것만 됨을 확인할 수 있다.

그러면 가능성은 뒷자리 2개의 수가 13, 23, 33, 43, 53, 63, 73, 83, 93 인가?로 줄어든다. 이 중 한 개를 찾으면 되는데, 같은 방법으로 계산을 하면 답을 찾을 수 있다.

[11-12] 양의 정수 .에 대하여, 1부터 .까지의 정수들 중에서 다음 조건을 모두 만족하도록 하나 이상을 뽑아 나열한 것을 KOI 수열이라고 하자.

(1) 두 번째 수부터는 바로 이전의 수보다 반드시 커야한다. (2) 홀수 번째 위치하는 수는 반드시 홀수여야 하고, 짝수 번째 위치하는 수는 반드시 짝수이어야 한다.

예를 들어, n = 3일 때 KOI 수열은 다음과 같은 네 가지가 있다.

1   3   1,2   1,2,3

11. n = 5일 때, KOI 수열은 몇 가지인가?

① 8   ② 10   ③ 12   ④ 15   ⑤ 18

** 출제의도?
논리적으로 따라가기?
** 의미?
주어진 규칙에 따라 정확히 만들 수 있는가?
** 해결예시?
논리적으로 경우를 나눠서 해결할 수 있다.
n=5 일 때,
한 개 선택 : 1, 3, 5
두 개 선택 : 1,2 1,4 3,4
세 개 선택 : 1,2,3 1,2,5 1,4,5 3,4,5
네 개 선택 : 1,2,3,4
다섯 개 선택 : 1,2,3,4,5

12. n = 10일 때, KOI 수열은 몇 가지인가?

① 140   ② 141   ③ 142   ④ 143   ⑤ 144

** 출제의도?
관계식 만들기
** 의미?
수열의 생성 규칙을 보고, 관계식/점화식을 만들어야 해결 할 수 있다.

10가지 경우를 모두 해 볼 수 없다.

** 해결예시?

즉, 11번에서 얻어낸 결과를 관찰을 해서 점화 관계식을 찾아내야 한다.

f(n) = f(n-2) + f(n-1) + 1

n=1 이면 f(1) = 1 <== 1

n=2 이면 f(2) = 2 ... <== 1, 1,2

13. 0과 1로 이루어진 길이가 6인 두 문자열 A, B에 대하여 A와 B의 거리는 첫 번째 위치부터 여섯 번째 위치까지의 각 위치에 대하여, 대응하는 두 문자가 서로 다른 위치의 개수이다. 예를 들어 A = 010010이고, B = 110101에 대하여 첫 번째, 네 번째, 다섯 번째, 여섯 번째 위치에서 A와 B의 대응하는 문자가 다르므로, A와 B의 거리는 4이다.

0과 1로 이루어진 길이가 6인 여러 문자열들의 모임에 대하여 이 모임의 유사도는 모든 두 문자열의 거리의 합이다. 예를 들어, 세 개의 문자열 A = 010010, B = 110101, C= 111000의 모임의 유사도 = A와 B의 거리 + A와 C의 거리 + B와 C의 거리 = 4 + 3 + 3 = 10이다.

길이가 6인 8개의 문자열 모임이 주어져 있다. 이 문자열 모임의 유사도를 구하기 위하여, 각 위치에 대하여 8개의 문자열에 있는 1의 개수를 세었더니 첫 번째 위치부터 여섯 번째 위치까지 차례대로 4, 3, 0, 7, 3, 2이었다. 그러면 이 문자열 모임의 유사도는 얼마인가?

① 57 ② 59 ③ 61 ④ 63 ⑤ 65

** 출제의도?

계산 변환

** 의미?

물론, 프로그래밍으로 모든 경우를 다 돌려볼 수 있지만, 생각을 바꾸면 계산을 쉽게 할 수 있다.

** 해결예시?

각 자리 1의 개수 4 3 0 7 3 2

각 자리 0의 개수 4 5 8 1 5 6

-> 각 자리끼리만 살펴보고 합치면 된다. 즉, 1인 경우는 0인 경우만큼 더하고, 0인 경우는 1인 경우만큼 더한다. 잘 보면 그냥 서로 곱해서 더하면 된다는 것을 알 수 있다.

각 자리 1의 개수 4 3 0 7 3 2

각 자리 0의 개수 4 5 8 1 5 6

곱 16+15+0+7+15+12

[14-15] 선생님께서 여섯 명의 학생들 A, B, C, D, E, F를 다음 조건을 만족하도록 한 줄로 세우는 과제를 내었다.

(1) 학생 A, B, C는 연속적으로 설 수 없다. 즉, 연속적으로 ABC, ACB, BAC, BCA,CAB, CBA로 설 수 없다.
(2) 학생 D, E, F는 연속적으로 설 수 없다. 즉, 연속적으로 DEF, DFE, EDF, EFD,FDE, FED로 설 수 없다.
(3) 학생 B는 제일 처음이나 제일 마지막에 설 수 없다.
(4) 학생 F는 B의 바로 앞이나 바로 뒤에 설 수 없다.
(5) 학생 A는 E보다 앞에 서야 한다.

14. 학생 E가 줄의 세 번째에 서게 되고, F가 네 번째 서게 된다면, 다음 중에서 맞는 것은?

① A는 줄의 두 번째에 선다.
② B는 줄의 다섯 번째에 선다.
③ C는 줄의 다섯 번째에 선다.
④ D는 줄의 첫 번째에 선다.
⑤ D는 줄의 두 번째에 선다.

** 출제의도?
논리적 사고
** 의미?
** 해결예시?
0 0 E F 0 0
0 B? E F B? 0 <- (3)에 의해서
0 B E F 0 0 <- (4)에 의해서
A B E F C D 확정 <- (5)+(2)에 의해서

15. 학생 C가 줄의 첫 번째에 서게 되고, A가 다섯 번째 서게 된다면, 다음 중에서 항상 옳 것은?

① D는 줄의 세 번째에 선다.
② F는 줄의 네 번째에 선다.
③ B는 A의 바로 앞에 선다.
④ C는 F의 바로 앞에 선다.
⑤ D는 B의 바로 앞에 선다.

** 출제의도?
논리적 사고
** 의미?
** 해결예시?
C 0 0 0 A 0
C 0 0 0 A E <- (5)에 의해서
C B? 0 F? A E <- (4)에 의해서
C F? 0 B? A E <- (4)에 의해서

결국 C X D X A E 을 알 수 있다.

※ [16-40] 문제나 프로그램 내에 명시되지 않은 모든 변수와 배열은 int 형이다.

16. 다음 프로그램의 출력 결과는 무엇인가?

```
int u=75, x=90, y=85;
int m;
m=u;
if(m<x) m=x;
if(m<y) m=y;
u=u+100-m;
x=x+100-m;
y=y+100-m;
printf("%d\n", y);
```

① 75 ② 85 ③ 90 ④ 95 ⑤ 100

** 출제의도?
코드의 이해
** 의미?
** 해결예시?
큰 값을 찾고 그냥 계산한다.

17. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, y=0;
for(i=2000; i<=2013; i++){
  if((i%4==0 && i%100!=0) || i%400==0) y++;
}
printf("%d\n",y);
```

① 4 ② 5 ③ 6 ④ 7 ⑤ 8

** 출제의도?
조건 실행 구조와 논리식의 이해
** 의미?
윤년 판단식을 조건 실행 구조와 논리식으로 구성한 것
** 해결예시?
2000 ~ 2013 까지로 몇 개 안되니 400으로 나누어지면 윤년, 4의 배수이면서 100으로 나누어 떨어지지 않으면 윤년이다.
참고 : 왜 이렇게 윤년을 계산하는가? 지구 공전 주기는 정확히 365일이 아니고

365.2564 일이다.
따라서 1년을 365일로 잡으면, 0.2564... 일씩 모자라게 된다.
그렇기 때문에 대략 4년 마다 하루를 늘려주면, 약 1.0256 일이 되기 때문에 적당히 보상된다.

18. 다음 프로그램의 출력 결과는 무엇인가?

```
int i=1, count=0;
while(i<=2013){
  count++;
  i=i*2;
}
printf("%d\n", count);
```

① 9 ② 10 ③ 11 ④ 12 ⑤ 13

** 출제의도?
코드 이해
** 의미?
2^n ??
** 해결예시?
1, 2, 4, 8, 16, ... , 2048

19. 다음 프로그램의 출력 결과는 무엇인가?

```
int sum=0, a=2013;
while(a>0){
  sum+=a%3;
  a/=3;
}
printf("%d\n", sum);
```

① 9 ② 10 ③ 11 ④ 12 ⑤ 13

** 출제의도?
코드의 이해
** 의미?
3진법 변환. 각 자리 수의 합
** 해결예시?
2013을 3진법으로 변환하면 2022120이 된다. 각 자리 숫자의 합을 구하면 된다.

20. 다음 프로그램의 출력 결과는 무엇인가?

```
int i;
int count=0;
for(i=1; i<125; i++){
  if(i%2==0 || i%5==0) count++;
}
printf("%d\n", count);
```

① 73 ② 74 ③ 75 ④ 76 ⑤ 77

** 출제의도?
코드의 이해
** 의미?
125보다 작은 2 또는 5의 배수의 개수
** 해결예시?
2의 배수 62개, 5의 배수 24개, 10의 배수 12개
이를 이용해 2또는 5의 배수의 개수를 구할 수 있다.

21. 다음 프로그램의 출력 결과는 무엇인가?

```
int i=1, ret=2;
while(i<2013){
  ret=ret+1;
  i=i+2013/ret;
}
printf("%d\n", ret);
```

① 4 ② 5 ③ 6 ④ 7 ⑤ 8

** 출제의도?
코드의 이해
** 의미?
** 해결예시?
단순 트레이싱

22. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[4]={4,3,1,2};
int i, j, k, t;
t=0;
for(i=0; i<4; i++){
  for(j=i+1; j<4; j++){
```

```
    if(a[j]<a[i]){
      k=a[j];
      a[j]=a[i];
      a[i]=k;
      t++;
    }
  }
}
printf("%d\n", t);
```

① 3　② 4　③ 5　④ 6　⑤ 10

** 출제의도?
코드의 이해
** 의미?
선택 정열의 이해. 자리 교환 횟수 출력한다.
** 해결예시?
4312
1432 -> 2번 교환
1243 -> 2번 교환

23. f(10)의 값은 무엇인가?

```
int f(int n){
  int i, j, ret=0;
  for(i=1; i<n; i++){
    for(j=1; j<2*i; j++) ret=ret+1;
  }
  return ret;
}
```

① 45　② 55　③ 81　④ 90　⑤ 100

** 출제의도?
함수의 이해. 이 함수가 어떤 짓을 하는 함수인가?
패턴 분석 및 이해, 목표에 대한 이해?
** 의미?
단순 함수에 대한 이해도 있지만, 함수의 패턴이 어떻게 만들어지는지에 대한 이해가 필요한 문제이다. 만들어지는 패턴을 이해하기 위해서 몇 개 해 본다.
무엇을 구하는 건지 살펴본다.
그 다음에 만들어지는 패턴에 대해서 구하는 것을 어떻게 빠르게 구할지 고민한다.
** 해결예시?
몇 개의 패턴을 만들어 보면
1　2　3　4　5　6　7　8 ...

```
1   1   1   1   ..
    2   2   2
    3   3   3
        4   4
        5   5
            6
            7
```

1부터 n까지.. 수가 만들어지는 패턴인데, 그 수의 개수의 총합이니, 1개, 3개, 5개, 7개, 9개..... 17개(9인 경우) 모두 더하면 된다. 더할 때에도 짝지어 더하면 빠르다

24. 배열 a에는 다음과 같은 값이 주어진다.

```
int a[]={1,37,3,10,13,23,4,8,17};
```

아래 주어진 함수 SomeAverage(a, 9, 5)를 호출했을 때 출력결과는 무엇인가?

```
void SomeAverage(int a[], int n, int m){
  int i, j, sum;
  for(i=0; i<=n-m; i++){
    sum=0;
    for(j=i; j<i+m; j++) sum=sum+a[j];
    printf("%d ", sum/m);
  }
  printf("\n");
}
```

① 13 17 11 12 13   ② 12 17 10 11 13
③ 12 17 11 12 13   ④ 12 17 10 12 13
⑤ 12.8 17.2 10.6 11.6 13

** 출제의도?
함수의 이해, 의미 분석
** 의미?
무엇을 하는 코드인가? 유심히 살펴보면, 배열을 전달받아서 5개씩 부분합을 만들고, 그 평균(정수/정수 잘림)을 출력하는 함수이다.
패턴을 찾기 위해 처음 몇 개를 직접 손으로 그려보면 된다.
** 해결예시?

```
          1 37 3 10 13  23 4 8 17    <- 9개의 원소
5개합               64
                       86 <- 모두 더할 필요 없이. 64에서 1빼고, 23 더하면 된다. 변화만 보면 된다.
                         53
                           58
                             65
```

5로 나눈 평균(정수) 이니 순서대로 12, 17, 10, 11, 13

25. f(2013)의 값은 무엇인가?

```
int f(int n)
{
  int a[2014];
  int i, j, cnt;
  for(i=1; i<=n; i++) a[i]=0;
  for(i=1; i<=n; i++){
    for(j=i; j<=n; j+=i) a[j]=1-a[j];
  }
  cnt=0;
  for(i=1; i<=n; i++){
   if(a[i]==1) cnt++;
  }
  return cnt;
}
```

① 10 ② 11 ③ 22 ④ 33 ⑤ 44

** 출제의도?
코드의 이해, 함수의 이해, 패턴분석, 의미분석
** 의미?
2013 이하의 완전제곱수 찾기(약수의 개수가 홀 수)이다
** 해결예시?
루트(2013) 44.???? 따라서 완전 제곱수는 44개이다.

26. f(25, 60)을 실행한 후, count의 값은 무엇인가?

```
int count=0;
int f(int a, int b){
  count++;
  if(a==b) return a;
  else{
    int mid;
    mid=(a+b)/2;
    return f(a, mid)+f(mid + 1, b);
  }
}
```

① 67 ② 68 ③ 69 ④ 70 ⑤ 71

** 출제의도?
재귀 함수의 이해

** 의미?
2^n 으로 갈라지는 재귀 함수 호출에 대한 이해
몇 번 갈라지는가?
f(a, b) 의 문제는 보다 작은 문제들의 합 ==> f(a, b) = f(a, mid) + f(mid+1, b)
결국에는 2^n 으로 갈라진 재귀 함수에 의한 2진 트리의 단말 노드의 개수를 출력하는 의미가 된다.
** 해결예시?
계속 분해 한 후 더하기

27. f(6, 3)의 값은 무엇인가?

```
int f(int n, int k){
  if(k<=0 || k>=n) return 1;
  return f(n−1, k−1)+f(n−1, k)+f(n−1, k+1);
}
```

① 53 ② 57 ③ 71 ④ 29 ⑤ 23

** 출제의도?
재귀함수의 이해, 백트랙킹과 다이나믹 프로그래밍.
** 의미?
26번처럼 분기해 나가려고 보면 3개로 분기되는 재귀함수이기 때문에 3^6개 이상 그려야 한다.
하지만, 재귀함수에 의한 백트랙킹은 거꾸로 표를 채워 올라가는 다이나믹 테이블 채우기가 가능하다는 것이 핵심이다.
** 해결예시?
표로 변환해 바꿔 그리고 주어진 관계를 이용해 표의 내용을 채워 나간다.

k		0	1	2	3	4	5	6	7	8	9	...	
n	0	1	1	1	1	1	1	1	1	1	1	...	<- 가장 처음 상태
	1	1	1	1	1	1	1	1	1	1	1	...	
	2	1	3	1	1	1	1	1	1	1	1	...	
	3	1	5	5	1	1	1	1	1	1	1	...	
	4	1	11	11	7	1	1	1	1	1	1	...	
	5	1	23	29	19	9	1	1	1	1	1	...	
	6	1	53	...			57...						

<- k>=n 인 경우 1,왼쪽 3은 윗줄 3개를 합친 값 좌상 f(n-1,k-1), 상 f(n-1, k), 우상 f(n-1, k+1)
표를 만들어서 채워 나가면 된다. 백트랙킹은 이렇게 표채우기 다이나믹 테이블로 쉽고 빠르게 가능하다. 자 그러면 f(6, 3)은? 참고로 f(5,1)은 위의 표에서 보면 23이다.

28. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[20]={1,4,3,2,4,1,2,5,1,2,3,5,4,1,2,1,1,1,1,9};
int i, s=0, gr=0;
for(i=0; i<20; i++){
 if(a[i]+s>10){
   gr++;
   s=0;
  }
  else s=s+a[i];
}
if(s>0){
  s=0;
  gr++;
}
printf("%d\n", gr);
```

① 1 ② 4 ③ 5 ④ 6 ⑤ 20

** 출제의도?

코드의 이해

** 의미?

별다른 의미는 없는 것으로 보이지만, 계속 더해가다가 일정 크기를 넘으면 카운팅하는 코드이다.

** 해결예시?

처음에 합 s를 0으로 놓고, 더해가기 시작한다. 그러다가 10을 넘으면 카운팅 하고, 합 s를 0으로 리셋 한다.

```
1 4 3  2  4 1 2 5 1  2  3 5  4  1 2 1 1 1 1 9
1 5 8 10 14   <-- 카운팅
            1 3 8 9 12  <-- 카운팅
                      3 8 12  <-- 카운팅
                             1 3 4 5 6 7 16 <-- 카운팅
```

29. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[4][4]={0};
int i, j, n, u, v, k, t;
i=j=0;
u=0, v=1;
n=4;
for(k=1; k<=16; k++){
  a[i][j]=k;
  if(i+u>=n || j+v>=n || i+u<0 || j+v<0 || a[i+u][j+v]!=0){
```

```
    t=v;
    v=-u;
    u=t;
  }
  i=i+u, j=j+v;
}
printf("%d\n", a[2][2]);
```

① 12　② 13　③ 14　④ 15　⑤ 16

** 출제의도?
코드의 이해
** 의미?
표 숫자 채우기. 달팽이 채우기.
** 해결예시?
표가 작으므로 따라가면서 그려본다.

1	2	3	4
12	13	14	5
11	16	15	6
10	9	8	7

30. 다음 보기 중에서 가장 큰 값은 무엇인가?

```
int f(int n, int m)
{
  if(n==0 && m==0) return 0;
  if(n%2==1 && m%2==1) return f(n/2, m/2)+1;
  if(n%2==0 && m%2==0) return f(n/2, m/2)+1;
  return f(n/2, m/2);
}
```

① f(18, 19)　② f(11, 25)
③ f(2, 17)　④ f(25, 16)
⑤ f(11, 29)

** 출제의도?
재귀 함수의 이해?
** 의미?
n, m 의 쌍이 둘 다 홀수이거나 둘 다 짝수이면 카운팅 한다.
** 해결예시?
f(11, 25)는 3으로 계산된다.

31. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[4]={0,0,0,0}, b[4]={6,2,1,1};
int k, n, i;
k=14;
for(n=0; n<4; n++){
  for(i=0; i<4; i++){
    if(a[i]==0){
      if(b[n]<k){
        k-=b[n];
      }
      else{
        a[i]=1;
        printf("%d ", i);
        break;
      }
    }
  }
}
printf("\n");
```

① 2 0 1 3 ② 2 0 3 1 ③ 2 1 0 3 ④ 3 1 4 2 ⑤ 3 2 1 4

** 출제의도?
  코드의 이해
** 의미?
  단순 트레이싱
** 해결예시?

32. 다음 프로그램의 출력 결과는 무엇인가?

```
int c=0, i, j, k;
char str[]="mississippi";
int len=11;
int p;
for(i=0; i<len; i++){
  for(j=i+1; j<len; j++){
    p=j-i+1;
    for(k=0; i+k<j-k; k++){
      if(str[i+k]!=str[j-k]){
        p=0;
```

```
          break;
        }
      }
      c = c + p;
    }
}
printf("%d\n", c);
```

① 11 ② 27 ③ 33 ④ 35 ⑤ 44

** 출제의도?
코드의 이해
** 의미?
회문(펠린드롬)의 길이.
각 문자에서 시작할 수 있는 회문의 길이를 계산하고 있다.
** 해결예시?
예를 들어 두 번째 i 에서는 issi, ississi 이렇게 길이가 4, 7인 것을 만들 수 있다.
그럼 세 번째 s 에서는 어떻게 될까?

33. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[8][8]={
  {0,1,1,1,1,0,0,0},
  {1,1,0,0,0,1,1,1},
  {0,0,1,1,1,0,0,1},
  {0,0,1,0,1,0,1,1},
  {0,0,1,1,1,0,1,0},
  {0,0,0,0,1,1,1,0},
  {0,1,1,0,0,0,0,0},
  {0,1,1,0,0,1,1,1}
};
void f(int n, int m){
  if(n<0 || m<0 || n>=8 || m>=8) return;
  if(a[n][m]!= 1) return;
  a[n][m]=2;
  f(n-1, m);
  f(n, m-1);
  f(n+1, m);
  f(n, m+1);
}
int main(){
```

```
  int i, j, g;
  g=0;
  for(i=0; i<8; i++){
    for(j=0; j<8; j++){
      if(a[i][j]==1){
        f(i, j);
        g=g+1;
      }
    }
  }
  printf("%d\n", g);
  return 0;
}
```

① 3 ② 4 ③ 5 ④ 30 ⑤ 31

** 출제의도?
코드의 이해, 플러드 필(flood fill)
** 의미?
4방향으로 펼쳐지는 재귀함수에 의해 붙어있는 1의 덩어리 개수를 계산할 수 있다.
얼룩 갯수 구하기. 4방향으로 지나가는 곳에는 2로 칠해둔다.
다시 1이 있는 곳에서는 다시 4방향으로 재귀 탐색해서 채우기 한다.
** 해결예시?
1의 덩어리 개수는 4이다.

34. f(10)의 값은 무엇인가?

```
int f(int n){
  int p[11][11];
  int i, j, ret;
  for(i=0; i<=n; i++){
    for(j=0; j<=n; j++) p[i][j]=0;
  }
  p[0][0]=1;
  for(i=0; i<=n; i++){
    p[i][0]=1;
    for(j=1; j<=i; j++){
     p[i][j]=p[i-1][j-1]+p[i-1][j];
    }
  }
  ret=0;
```

```
  for(i=0; i<=n; i++) ret=ret+p[n][i];
  return ret;
}
```

① 511 ② 512 ③ 513 ④ 1023 ⑤ 1024

** 출제의도?
2차원 다이나믹 테이블 채우기
** 의미?
파스칼 삼각형 채우기, 점화 관계식을 보고 표의 수를 채워나가면 된다.
** 해결예시?

1 0 0 0 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0 0 0
1 2 1 0 0 0 0 0 0 0 0
1 3 3 1 0 0 0 0 0 0 0
…

35. 다음 보기 중에서 가장 큰 값은 무엇인가?

```
int f(int n)
{
  int a[10]={0}, t=0, i;
  while(n>0){
    a[n%10]=1;
    n=n/10;
 }
 for(i=0; i<10; i++) t=t+a[i]*i;
 return t;
}
```

① f(711587) ② f(999999) ③ f(614331)
④ f(843211) ⑤ f(914312)

** 출제의도?
함수의 이해. 의미 파악
** 의미?
함수가 어떤 작업을 하는가? 를 파악하는 것이 중요하다.

함수의 내용을 보면 각 자리의 숫자들을 찾아내 체크하고, 체크한 수를 합하는 함수이다.
결국 들어온 수의 각 자리에 사용된 숫자들만 골라내서 그 합을 만들어내는 함수이다.
** 해결예시?
f(999999) 의 결과는 9
f(711587) 의 결과는 1 5 7 8 을 모두 더하면 21이 된다.

36. 아래 주어진 함수 AmicableNumbers()를 호출했을 때 출력 결과는 무엇인가?

```
int AN_helper(int num)
{
  int i, tot=0;
  for(i=1; i<num; i++){
    if(num%i==0) tot=tot+i;
  }
  return tot;
}
void AmicableNumbers(){
  int i, num;
  for(i=1; i<1000; i++){
    num=AN_helper(i);
    if(i<num && i==AN_helper(num)) printf("%d %d\n", i, num);
  }
}
```

① 100 117 ② 120 240 ③ 127 128
④ 220 284 ⑤ 496 496

** 출제의도?
함수의 이해
** 의미?
자기 자신을 제외한 약수의 합으로 서로 비교한다.
'진약수의 합이 서로를 나타내는 수' 의 의미로 이름 붙여진 친화수이다.
** 해결예시?
1000보다 작은 친화수는 220 284 이다.

37. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[10]={3456,7,8976,54,53,456,7865,45,6734,53};
int i, b, cnt=0;
for(i=0; i<10; i++){
  b=(a[i]&1) | (a[i]&2);
  if(b==0) cnt++;
}
printf("%d\n", cnt);
```

① 0 ② 1 ③ 2 ④ 3 ⑤ 4

** 출제의도?

코드의 이해, 비트단위 연산자의 이해

** 의미?

b=(a[i]&1) | (a[i]&2); //b의 값이 0이려면 2진수로 바꿨을 때 01이나 10이 아니어야 한다.

결국, 4의 배수(2진수로 바꾸었을 때 ...00 인 형태) 찾기 문제가 된다.

** 해결예시?

4의 배수인 것만 고르면 된다.

38. 다음 프로그램의 출력 결과에 포함되지 않는 것은 무엇인가?

```
char l[10];
void f(int r, int s, int len){
  if(r==0 && s==0) printf("%s\n", l);
  if(r>0){
    l[len]='i';
    f(r-1, s+1, len+1);
  }
  if(s>0){
    l[len]='o';
    f(r, s-1, len+1);
  }
}
int main()
{
  int i;
  for(i=0; i<10; i++) l[i]='\0';
  f(4, 0, 0);
  return 0;
}
```

① iioooiio ② ioioioio ③ iiiioooo

④ iioiiooo ⑤ ioiioioo

** 출제의도?

재귀함수의 이해? 카탈란 수?

** 의미?

2개로 퍼져나가는 재귀함수의 특징을 잘 보면

왼쪽은.. (-, +) 오른쪽은 (0, -) 형태로 퍼진다.

왼쪽은 i가 붙어가고 오른쪽은 o가 붙는데,

재귀함수의 형태상

왼쪽에 붙은 i의 개수는 반드시 o의 개수와 같거나 보다 많아야 한다.

** 해결예시?

직접 그려봐도 된다.

* 실기 대회 연습 문제(코드 해석만 첨가)

[39-40] 다음과 같은 문제를 해결하기 위해 프로그램을 작성하였다. 물음에 답하시오.

KOI 장거리 달리기 대회가 진행되어 모든 선수가 반환점을 넘었다. 각 선수의 입장에서 자기보다 앞에 달리고 있는 선수들 중 평소 실력이 자기보다 좋은 선수를 남은 거리 동안 앞지르는 것은 불가능하다. 반대로, 평소 실력이 자기보다 좋지 않은 선수가 앞에 달리고 있으면 남은 거리 동안 앞지르는 것이 가능하다. 이러한 가정 하에서 각 선수는 자신이 앞으로 얻을 수 있는 최선의 등수를 알 수 있다.

각 선수의 평소 실력은 정수로 주어지는데 더 큰 값이 더 좋은 실력을 의미한다. 현재 달리고 있는 선수를 앞에서 부터 표시했을 때 평소 실력이 각각 2, 8, 10, 7, 1, 9, 4, 15 라고 하면 각 선수가 얻을 수 있는 최선의 등수는 (같은 순서로) 각각 1, 1, 1, 3, 5, 2, 5, 1 이 된다. 예를 들어, 4 번째로 달리고 있는 평소 실력이 7 인 선수는 그 앞에서 달리고 있는 선수들 중 평소 실력이 2 인 선수만 앞지르는 것이 가능하고 평소실력이 8 과 10 인 선수들은 앞지르는 것이 불가능하므로, 최선의 등수는 3 등이 된다.

선수들의 평소 실력을 현재 달리고 있는 순서대로 입력 받아서 각 선수의 최선의 등수를 계산하는 프로그램을 작성하시오.

입력 형식

입력 파일의 이름은 INPUT.TXT 이다. 첫째 줄에는 선수의 수를 의미하는 정수 N 이 주어진다. N 은 3 이상 500,000 이하이다. 다음 줄에는 N 개의 정수가 주어진다. 이 값들은 각 선수들의 평소 실력을 앞에서 달리고 있는 선수부터 제시한 것이다. 각 정수는 1 이상1,000,000,000 이하이다. 단, 참가한 선수들의 평소 실력은 모두 다르다.

출력 형식

출력 파일의 이름은 OUTPUT.TXT 이다.

각 선수의 최선의 등수를 나타내는 정수 N 개를 입력에 주어진 선수 순서와 동일한 순서로 한 줄에 출력한다.

모든 정수들 사이에는 하나의 공백을 둔다.

입력과 출력의 예

입력(INPUT.TXT)

```
8
2 8 10 7 1 9 4 15
```

출력(OUTPUT.TXT)

```
1 1 1 3 5 2 5 1
```

프로그램

```
#include <stdio.h>
#define MAXN 500000
//데이터의 개수가 50만개이다. 이런경우 일반적인 O(n^2) 방법으로는 시간 초
  과 발생한다.
//각각에 대해서 왼쪽 것을 계속 비교해서 카운트하면 그 상황이 O(n^2) 상황이
  다.
//그래서 재귀함수를 활용한 분할 후, 병합정렬 아이디어로 O(n log2 n)으로 낮
  추는 코드?이다.

//값, 인덱스, 답 을 위한 구조체 형태 선언
struct Info { int value, index, answer;};

//그 구조체 형태로 배열 잡음. temp 배열은 임시용
struct Info info[MAXN], temp[MAXN];

//최대로 가능한 등수를 기록하기 위한 배열
int n, answer[MAXN];

//범위를 입력 받아 뭔가 처리한다. 2중 재귀로 계산량이 O(n log2 n) 로 1초
  내에 해결?
void merge(int left, int right){

//범위가 1개 이하가 되면 재귀 호출 중지, 최소 2개~3개로 쪼개짐?
  if(left>=right) return;
  int i, l, r, mid;

//범위의 가운데 잡고
  mid=(left+right)/2;

//왼쪽에서 가운데까지 같은 방법으로...
  merge(left, mid);

//가운데에서 오른쪽까지 같은 방법으로...
```

```
  merge(mid+1, right);

//가운데를 기준으로 왼쪽 부분의 첫번째를 l 오른쪽 부분의 첫번째를 r 로 셋
  팅
  l=left, r=mid+1;
  for(i=left; i<=right; i++){

//이런 경우이면
//왼쪽 그룹 것 정보 복사하고 그 오른쪽 것으로 넘어감...
    if ( (a)  || ( (b)  && (c) )) temp[i]=info[l++];
    else{

//오른쪽 그룹 것 정보 복사하고, 그 오른쪽 것으로 넘어감
      temp[i]=info[r++];

//가능한 등수는 이전에 얻었던 등수에 현재 집합에서의 등수 더하기?
      temp[i].answer+=mid-l+1;
    }
  }

//임시로 만들어낸 구조체 배열 temp[ ] 를 info[ ] 배열에 복사
  for(i=left; i<=right; i++) info[i]=temp[i];
}
int main(){
  int i;
//입력 파일 설정
  FILE *fi, *fo;
  fi=fopen("INPUT.TXT", "r");

//입력 파일 첫번째 정수 읽기
  fscanf(fi, "%d", &n);

//입력 파일에서 한 줄 읽기
  for(i=0; i<n; i++){

//읽어서.. 값을 넣고
   fscanf(fi, "%d", &info[i].value);

//읽어들인 순서 정보를 info[].index 에 기록
```

```
    info[i].index=i;
   }

//파일 입력 끝. 파일 닫기
  fclose(fi);

//재귀 함수 호출
  merge(0, n-1);

//읽어들인 순서대로 최대로 가능한 등수는, 그 값의 answer
  for(i=0; i<n; i++) answer[info[i].index]=info[i].answer;
  fo=fopen("OUTPUT.TXT", "w");

//answer 배열에 저장된 값에 1더해서 등수 출력
  for(i=0; i<n; i++) fprintf(fo, "%d ", answer[i]+1);
  fclose(fo);
  return 0;
}
```

39. 다음 중 빈칸 (a), (b), (c)에 들어갈 내용으로 알맞게 짝지어진 것은?

① (a) r > right	(b) l <= mid	(c) info[l].value > info[r].value
② (a) r >= right	(b) l < mid	(c) info[l].value < info[r].value
③ (a) r > right	(b) l <= mid	(c) info[l].value < info[r].value
④ (a) l > mid	(b) r <= right	(c) info[l].value > info[r].value
⑤ (a) l > mid	(b) r <= right	(c) info[l].value < info[r].value

40. 입력 파일이 다음과 같이 주어졌을 때, 위 프로그램의 출력은 무엇인가?

```
10
3 12 16 1 10 2 11 7 15 13
```

① 1 1 1 4 3 4 3 5 2 3
② 1 1 1 3 3 5 3 5 2 3
③ 1 1 1 4 3 5 4 5 2 3
④ 1 1 1 4 3 5 3 5 1 3
⑤ 1 1 1 4 3 5 3 5 2 3

2014 시도예선 초등부 문제

1. (7 + 77 + 777 + 7777 + 77777)÷7의 값은?

① 12341 ②12342 ③ 12343 ④ 12344 ⑤ 12345

** 출제의도?

패턴 분석, 계산 단순화.

** 의미?

계산식의 패턴을 보고 보다 간단한 계산식으로 변형하여 계산 시간 줄이기

** 해결예시?

7로 먼저 나누면 (1+11+111+1111+11111)가 되고, 앞자리부터 각 자리 당 1씩 증가한다.

2. 아래처럼 세 자리 자연수 ABC와 DEF를 더해서 1000이 되었다. 여기서, A, B, C, D, E, F는 모두 0이 아니다. 그러면 A+B+C+D+E+F는 얼마인가?

```
   A B C
 + D E F
-----------
 1 0 0 0
```

① 28 ② 29 ③ 30 ④ 31 ⑤ 32

** 출제의도?

복면산(문자나 기호로 대치된 숫자 계산, 복면(마스크) 계산), 특징 점 찾기, 목표 값과의 관계 생각

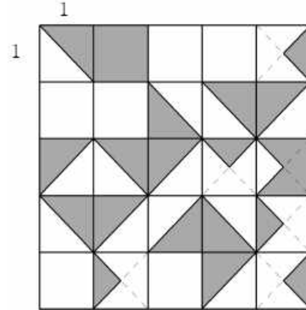
** 의미?

주어진 상황에서 특징 점을 찾고, 그 특징 점을 이용해 목표 값을 계산하기.

** 해결예시?

미지수가 n개이고, 서로 다른 방정식이 n개 보다 적으면, 비례식 또는 불능식이 될 수 있다. 따라서 주어진 복면산에서 어떤 특징 점을 찾아내고 그 특성을 이용해 목표값을 계산한다. 주어진 복면산 식을 보고 미지수를 찾아내기 위한 계산식을 만들어낼 수 있지만, 주어진 식을 관찰하고 목표값을 보면 쉽게 해결된다. 두 수의 합의 1의 자리(마지막자리)가 0이 되려면, 한 자리 숫자의 합이 0으로 끝나야 하는데. 어떤 경우도 20을 만들 수 없다. 따라서 두 수 C, F의 합은 그 경우가 어떤 것이든지 간에 10이 되어야 한다. 다시 10의 자리로 올라가보면, 아래서 자리올림 된 1이 있으니, 마찬가지 논리로 B+E=9가 되어야 한다. A+D도 마찬가지로 9이다. 적당한 관계식을 찾았고, 목표값은 A+B+C+D+E+F 이다. 그렇다면 얼마일까?

3. 변이 가로와 세로인 가장 작은 정사각형들은 모두 변의 길이가 1이다. 색칠된 영역의 면적은 얼마인가?



① 7 ② 7.5 ③ 8.5 ④ 9 ⑤ 9.25

** 출제의도?

패턴 분석, 빠르면서도 정확한 계산 아이디어

** 의미?

모든 것을 포함할 수 있는 단위 패턴을 찾고, 그 단위 패턴을 이용해 전체를 해결하는 방법이다. 분할 정복과 비슷하다. 그냥 일일이 계산하면 실수의 가능성이 매우 높으므로 실수를 하지 않으면서도 정확한 계산을 하기 위한 아이디어가 필요하다.

** 해결예시?

색이 칠해져있는 영역들을 살펴보면 1/4 단위 삼각형으로 다 나눌 수 있다. 따라서 색이 칠해진 각 사각형에 그런 단위 삼각형이 몇 개 들어갈 수 있는지만 순서대로 모두 센 후, 그 값에 1/4치면 된다.

4. 사과 a개와 바나나 b개가 들어있는 자루가 있다. 이 자루에 사과 3개를 추가로 넣었더니, 자루에 들어 있는 사과의 개수가 정확히 자루에 들어 있는 전체 과일 개수의 가 2/3가 되었다. 그렇다면 a > b를 만족하는 가장 작은 a값은?

① 1 ② 3 ③ 5 ④ 7 ⑤ 8

** 출제의도?

문제/상황에 대한 관계식 만들기, 관계식 해결

** 의미?

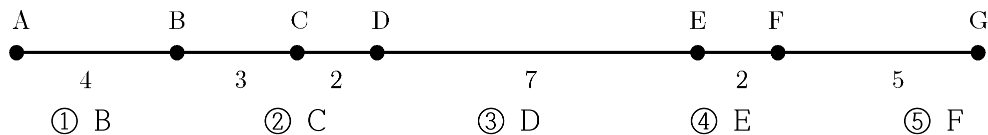
문제해결에서 중요한 것은 어떠한 문제/상황에서 찾아낼 수 있는 특징이나 관계를 찾아낸 후, 그것을 컴퓨터를 이용해 해결할 수 있도록 적당한 방법으로 표현하고, 그렇게 표현/저장된 데이터들을 이용해 유용한 결과들을 얻어내거나 분석하는 것이다.

따라서 문제/상황에서 관계나 특성을 찾아내고, 그 것을 이용해 원하는 결과를 얻어낼 수 있어야 한다.

** 해결예시?

문제/상황에서 관계들을 찾아내 방정식으로 표현하고, 해석해서 원하는 값을 찾아내면 된다. 주어진 문제에서 (a+3)=2/3((a+3)+b), a>b. 이므로, a=2b-3 이면서 a>b를 만족하는 가장 작은 자연수 a를 찾아내면 된다.

5. 아래 그림처럼 직선 도로에 집이 A, B, C, D, E, F, G 일곱 채가 있다. 이웃한 두 집 사이의 거리는 두 집 사이에 숫자로 표시되어 있다. 반상회를 하기 위해 한 집에 모이고 싶다. 이왕이면, 각 집에서 반상회가 열리는 집까지의 거리의 합이 가장 작았으면 한다. 어느 집에서 모여야 할까?



① B　　② C　　③ D　　④ E　　⑤ F

** 출제의도?

효율적인 계산. 이전 계산 결과 재사용하기.

** 의미?

프로그래밍 언어를 활용하는 문제해결에서는 정확한 결과와 함께, 보다 적은 계산량(계산시간)으로 가능하도록 하기 위한 아이디어와 방법을 필요로 한다. 계산량을 줄이기 위한 기본적인 방법은 한 번 계산했던 결과를 이용하는 것이다.

주어진 문제를 해결하기 위해 모든 경우에 대해서 다 다시 계산할 수 있지만, 한 번 계산한 결과와, 다시? 계산할 결과의 차이를 분석해 그 차이만큼만 계산한다면, 보다 빠른 계산이 가능하다.

** 해결예시?

A점에서 모이는 경우 : B, C, D, E, F, G 가 움직여 와야 한다. 거리의 합은 4, 4+3, 4+3+2, 4+3+2+7, 4+3+2+7+2, 4+3+2+7+2+5 로 계산할 수 있는데, 늘어나는 거리만큼 만 더해 가면 4, 7, 9, 16, 18, 23 의 합 인 77임을 계산할 수 있다.

B점에서 모이는 경우? : A점까지 77만큼 필요한데, A는 4만큼 더 이동해야하고, 나머지 6명은 4만큼 덜 이동하면 되므로 77 + 1*4 + (6*(-4)) = 59 로 계산할 수 있다.

C점에서 모이는 경우? : B점까지 59만큼 필요한데, A와 B가 3만큼씩 더 이동해야하고, 나머지 5명은 3만큼 덜 이동하면 되므로 59+2*3+(5*(-3)) = 50

D점에서 모이는 경우? : C점까지 50. 50+3*2+(4*(-2)) = 48

E점에서 모이는 경우? : D점까지 48. 48+4*7+(3*(-7)) = 55

F점. G점으로 가면 더 늘어난다.

6. 두 자연수 a와 b에 대하여 ab는 a를 b번 곱해서 얻은 수이다. 예를 들어, 43은 4*4*4와 같다. 1 + 22 + 33 + 44 + 55 + 66 + 77 + 88 + 99 + 1010 의 마지막 자리의 숫자(1의 자릿수)는 얼마인가?

① 1　　② 3　　③ 5　　④ 7　　⑤ 9

** 출제의도?

패턴 분석?, 목표 분석?

** 의미?

** 해결예시?

얻고자하는 답이 1의 자리의 숫자이므로, 각 제곱 승들을 모두 계산할 필요 없이 1의 자리만 고려해 더하고 확인하면 된다. 1+4+7+6+5+6+3+6+9+0 의 결과 값에서 1의 자릿수이다.

7. 동전이 10개씩 들어 있는 자루가 10개 있다. 이 중에 9개의 자루에는 모두 정상인 동전이 들어 있고, 나머지 하나의 자루에 들어 있는 동전은 모두 가짜이다. 정상인 동전 하나의 무게는 10g이고, 가짜 동전 하나의 무게는 9g이다. 무게를 그램(g) 단위로 잴 수 있는 저울을 최대 k번 이용하여 항상 가짜 동전이 들어 있는 자루를 찾을 수 있어야 한다. 가능한 가장 작은 k는 얼마인가?

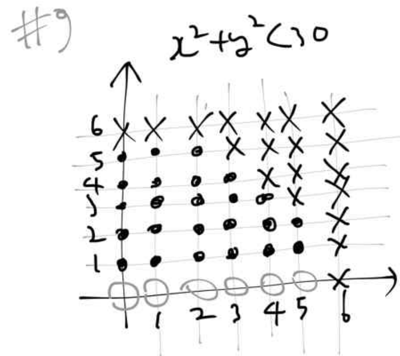
① 1 ② 2 ③ 3 ④ 5 ⑤ 7

** 출제의도?
퍼즐 해결 아이디어.
** 의미?
많이 알려져 있는 문제. 양팔 저울이 아니고 무게를 잴 수 있는 저울이 주어짐. 해싱과 연관
** 해결예시?
순서대로 자루를 세워두고 1번~9번 자루라고 이름을 붙여두고. 각 자루에서 0개, 1개, 2개, ... (10-1)개를 꺼내 모은 후 무게를 1번 재면 된다. 만약 모든 자루에 들어있는 동전들이 모두 정상 동전이라면? 45개의 동전이 10g이므로 490g으로 나와야 한다. 그런데 무게를 잰 결과가 485g라면? 5g만큼 차이가 나는 것이므로
첫 번째 자루부터 1개, 2개, 3개, ... (10)개를 빼는 것과 차이가 없을까?

8. $x^2+y^2<n$을 만족하는 정수 쌍(x, y)는 n=2 인 경우 (0, 0), (0, 1), (1, 0), (-1, 0), (0, -1)로 모두 5개 이다. n=30 이면 $x^2+y^2<n$을 만족하는 정수 쌍 (x, y)는 모두 몇 개인가?

① 91 ② 95 ③ 97 ④ 99 ⑤ 101

** 출제의도?
경우의 수 구하기? 하지만, 문제 형태 바꿔서 풀기? 패턴 인식? 계산 간략화?
** 의미?
원의 방정식 형태로 주어진 부등식에서 부등식을 만족 시킬 수 있는 경우의 개수를 찾는 문제이다.
가능한 경우의 개수를 모두 찾는 문제들 중 하나라고 볼 수 있지만, 가능한 경우가 매우 많다. 이럴 경우 문제의 형태를 바꾸어 접근하는 방법을 사용할 수 있다.
** 해결예시?
2차원 그래프 형태로 바꾸어 포함되지 않는 경계만 찾아볼 수 있다. 또한, 가능한 모든 경우를 찾을 필요 없이 반복적 패턴을 생각해 그 패턴 중 하나만 계산하고 전체로 확대 시켜서 판단해 볼 수 있다. 부등식을 만족하는 한 사분면의 점은 24개 이다. (0, 0)을 놓치지 않도록 해야 한다.



9. 세 개의 알람시계 A, B, C가 있다. A는 10초 동안 울리고 30초 동안 멈추고, B는 15초 동안 울리고 45초 동안 멈추고, C는 20초 동안 울리고 60초 동안 멈추는 과정을 각각 반복한다. 오후 6시 정각에 A, B, C가 동시에 울리기 시작했다. 그날 오후 6시 59분 59초까지 세 개의 알람이 동시에 울리는 시간(초)의 합은 얼마인가?

① 100 ② 120 ③ 140 ④ 150 ⑤ 240

** 출제의도?

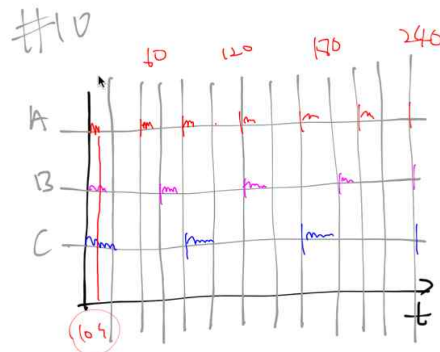
반복 패턴 찾기?

** 의미?

간트차트. 최소공배수(반복 구간) 찾기.

** 해결예시?

머릿속으로만 계산하면 헷갈리니 간트차트(gantt chart)를 그려보는데, 3개의 시계가 동시에 시작하는 주기(최소공배수)를 찾고 한 주기만 그린 후 3개의 시계가 모두 울리는 시간을 찾은 후 원하는 시간만큼 더하면 된다.



10. 다음은 하나의 직사각형을 9개의 작은 직사각형들로 나눈 도형이다. 이 도형에서 빗금 친 직사각형 안에 있는 숫자는 이 직사각형의 면적을 나타낸다. 전체 직사각형의 면적은 얼마인가?

12		24
	18	
16	12	

① 185 ② 188 ③ 190 ④ 192 ⑤ 195

** 출제의도?

가능한 경우의 수? 미지수 찾기? 정수론?

** 의미?

각 변의 길이를 미지수로 두고 시도해 볼 수도 있지만 가능한 여러 가지 중에서 논리적으로 맞는 값을 찾아야 한다. 그림을 보고 대충 비례적으로 짐작해서 찾을 수도 있지만, 이는 매우 정확하지 않은 방법이다.

** 해결예시?

각 부분 사각형의 넓이가 정수이므로, 각 구역의 길이는 모두 정수이다. 알려진 각 넓이를 만들 수 있는 약수들을 생각해 보면

12 : 1, 2, 3, 4

24 : 1, 2, 3, 4, 6, 8, 12

18 : 1, 2, 3, 6, 9

16 : 1, 2, 4, 8

인데, 16과 12는 한 변이 같기 때문에, 한 변은 1, 2, 4 중 하나가 되어야 한다. 각 경우에 대해서 다른 빗금 친 부분의 넓이를 계산할 수 있는지 논리를 확인해가면 각 변의 길이를 구할 수 있고, 사각형의 넓이를 구할 수 있다.

11. 6개의 수들이 4, 6, 2, 1, 3, 5의 순서로 나열되어 있다. 이 수들에 대하여, 임의의 두 수의 교환을 반복 수행하여, 작은 수부터 큰 순서대로 다시 나열하고자 한다. 교환하는 두 수는 인접하지 않아도 된다. 두 수의 교환 횟수의 최솟값은 얼마인가?

① 3 ② 4 ③ 5 ④ 6 ⑤ 7

** 출제의도?

선택 정렬?

** 의미?

어떤 집합에서 가장 작은 수들을 찾아 골라내면서 정렬하는 선택정렬 방법을 묻고 있는 문제이다.

** 해결예시?

[4, 6, 2, 1, 3, 5]에서 가장 작은 값(1)을 찾아 4와 자리를 바꾼다.

1, [6, 2, 4, 3, 5]에서 가장 작은 값(2)를 찾아 6과 자리를 바꾼다.

1, 2, [6, 4, 3, 5] ... 6과 자리를 바꾼다.

1, 2, 3, [4, 6, 5] ... 자리를 안 바꿔도 된다.

1, 2, 3, 4, [6, 5] ... 자리를 바꾼다.

1, 2, 3, 4, 5, 6 ... 완성

참고 : 가장 작은 값을 선택해가면 정렬하는 방법을 선택정렬이라고 하는데, 보통 2가지 형태로 알려져 있다. 한 가지는 집합에서 가장 작은 값의 위치를 찾아 마지막에 자리를 교환하는 방법이 있고, 한 가지는 비교를 계속하면서 그 때 그 때마다 자리를 교환하는 방법이다. 가장 작은 값을 선택한다는 의미에서 같은 선택정렬이지만, 데이터를 교환하는 횟수에서 차이가 있다.(비교횟수는 같다.) 위치를 찾은 후 마지막에 교환하는 방식은 데이터를 교환하는 횟수는 적지만, 실제 프로그래밍을 해보면 번거롭다. 반면에 그 때 그 때마다 값을 비교해서 교환하는 방식은 프로그래밍이 단순하다는 이점이 있다. 하지만, 실제 정렬에는 이런 선택 정렬은 매우 느린 정렬 방법에 속하기 때문에 거의 사용하지 않는다. 그냥 정렬 아이디어와 방법에 대한 간단한 예시 일 뿐 그 이상은 아무런 의미가 없다고 볼 수 있다.

참고2 : 정렬의 효율은 비교횟수와 교환횟수 둘 다 의미가 있지만, 일반적으로는 비교횟수로 비교하는 경우가 많다. 그렇다면, 좋은 정렬알고리즘이란? 정렬된 결과를 가정할 때, 각 데이터들을 그 위치로 최대한 빠르게 이동시킬 수 있는 방법이라고 할 수 있다. 어떻게 하면 자기가 가야할 자리로 가장 빠르게 보낼 수 있는지 그런 아이디어와 구체적 방법이 정렬 알고리즘의 효율성을 결정한다.

12. 축구 라이벌인 한국 축구팀과 일본 축구팀은 두 팀끼리만 몇 번의 시합을 했다. 비긴 시합은 없었다. 매 시합마다 이긴 팀은 A 그램의 메달을 받고 진 팀은 B 그램의 메달을 받는다. 항상 A > B이며, 둘 다 자연수이다. 한국이 받은 메달의 총 무게는 42그램이고, 일본이 받은 메달의 총 무게는 35그램이었다. 아는 사실 중 하나는, 일본은 정확히 2번의 시합만 이겼다는 것이다. 그렇다면 A + B의 값은 얼마인가?

① 7 ② 5 ③ 9 ④ 11 ⑤ 6

** 출제의도?
논리식 찾기. 구성.

** 의미?
주어진 문제/상황에서 논리식을 만들어내고 가능한 경우들에 대해 확인

** 해결예시?
문제/상황에서 주어진 사실들을 골라내 정리한다.

- n번의 시합, 비긴 시합은 없었다.
- 이기면 A그램 받고, 지면 B그램 받고, 항상 A>B, 둘 다 자연수.
- 한국이 받은 메달 무게 42그램, 일본이 받은 메달 무게 35그램.
- 일본은 2번만 이겼다.

전체 시합 횟수를 n이라고 생각하면, 한국은 n-2번을 이겼기 때문에 (n-2)*A+2*B=42, 일본은 2번 이겼기 때문에 2*A+(n-2)*B=35. 두 식을 서로 더하면 n*(A+B)=77 이기 때문에 n과 (A+B)는 1, 7, 11 중 하나씩이다. n은 1이 아니므로, 7또는 11이 되어야 하는데 n=7이면, (A+B)는 11이어야 하고, 이 경우 3*B가 13이 되어야 하는데, B가 자연수이기 때문에 안 된다.

13. 칠판에 연속된 여섯 개의 자연수가 적혀있다. 이 중에 하나의 수를 지우고 나머지를 더했더니, 2014였다. 지운 수는 무엇인가?

① 393　　② 395　　③ 397　　④ 399　　⑤ 401

** 출제의도?

문제/상황 분석/이해, 관계식(방정식) 찾기

** 의미?

주어진 문제/상황에 대한 관계식(방정식) 찾아 해결하기?

** 해결예시?

어떤 수로 시작할 지도 모르고, 그 중에서 어떤 수를 뺀 것인지도 알 수 없다. 그래서 연속된 자연수 중 가장 작은 수를 n이라고 하면,

n, n+1, n+2, n+3, n+4, n+5

라고 할 수 있는데, 한 수를 빼서 가장 큰 합은 (n+1)+(n+2)+(n+3)+(n+4)+(n+5), 한 수를 빼서 가장 작은 합은 n+(n+1)+(n+2)+(n+3)+(n+4) 이고, 합이 2014라고 하면 다음과 같은 부등식(관계식)을 만들 수 있다.

n+(n+1)+(n+2)+(n+3)+(n+4) <= 2014 <= (n+1)+(n+2)+(n+3)+(n+4)+(n+5)

5*n+10 <= 2014 <= 5*n+15 부등식을 양쪽을 나눠서 따로 계산하고 결합하면

399.8 <= n <= 400.8 이고 n은 자연수 이므로 n은 400이 된다.

400, 401, 402, 403, 404, 405 중에서 하나를 빼고 나머지를 모두 더해 14가 나오려면?

14. 두 변수 a와 b에 대해, 오로지 a = a + b 연산과 b = b + a 연산만 사용할 수 있다. 두 변수 a, b의 초기 정수 값 두 개와 하나의 정수 K가 주어지면, 두 연산을 원하는 횟수만큼 적용하여 두 변수 중 하나에 정수 K가 저장되도록 만들 수 있는 경우와 만들 수 없는 경우가 있다. 만약 a = 7이고 b = 5인 경우, 만들 수 없는 K 값은 무엇인가?

① 26　　② 32　　③ 33　　④ 35　　⑤ 37

** 출제의도?

대입 연산의 이해? 상태변화 경우의 수?

** 의미?

어떤 초기 조건에서 시작해서 변화 가능한 많은 경우들에 대해서 확인하고 특정 값을 찾아야하는 문제이다. 대입연산, 즉 어떤 값이 만들어지느냐 안 만들어지느냐에 대한 결정 문제이다. 추측 및 대입(주어진 선택 답안을 대입해서 확인) 불가능하다. 가능한 변화 상태가 아주 많다. 무계획적인 방법 또는 시행착오 방법으로는 해결이 불가능하다.

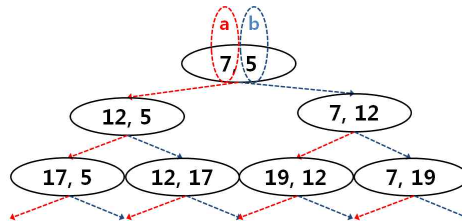
** 해결예시?

일단, a 이던지 b이던지 구분할 필요가 없다는 것을 생각해야 한다. 만들 수 있는지 없는지 이기 때문에 그 값이 a 이던지 b이던지 구분할 필요가 없다.

해결할 수 있는 다양한 시도와 방법이 있을 수 있다.

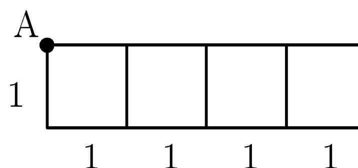
수학적 표현과 사고로 가능함. 하지만, 변화하는 상태 값만 주의 깊게 살펴보고 유추하는 방법으로도 매우 쉽게 가능하다. 상태 변화를 트리 구조로 표현해 찾는 방법도 있지만, 중복이 많이 생기기 때문에 시간이 오래 걸린다. 가능한 상태만 차곡차곡 키워가는 방법이 가

장 효과적이다.
첫 번째, 수학적 접근
값이 변하는 관계를 생각해 보면 적당히 곱의 합 형태로 나타나야 하는데 그 관계는 m*a+n*b 형태로 표현할 수 있다. 이 의미는 a의 1배, 2배, 3배, ... 에 b의 1배, 2배, 3배의 합으로 값이 만들어진다는 것을 의미한다. a의 초기값이 7, b의 초기 값이 5이고, 주어진 선택 답안 중에 최대 37이기 때문에, 기본으로 7, 5에 m의 최댓값은 6 정도까지, 거꾸로 n의 최댓값은 8 정도까지 생각할 수 있기 때문에 가능한 경우의 수를 만들어보면서 찾아 볼 수도 있다.
두 번째, 가능한 상태 변화를 트리 구조로 그려보는 방법
a의 값을 왼쪽에, b의 값을 오른쪽에 그려서 원으로 묶은 형태(노드, 정점)를 생각해 보면, 만들어지는 값들은 a에 b값을 더하는 경우 또는, b에 a값을 더해 변하는 2가지 경우뿐이다. 이를 그림(2진 트리)으로 표현하면 다음과 같다.



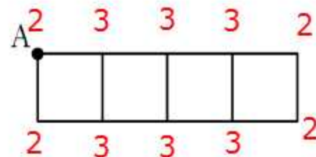
이렇게 찾아가면 반드시 찾아낼 수는 있다... 하지만.. 한 단계만 더 내려가도 8개.. 그 아래 더 내려가면 16개를 더 그려야 한다. 반드시 답을 찾아낼 수는 있지만, 중복이 많아서 시간이 매우 많이 걸릴 것이다. 이러한 형태의 상태 변화는 프로그래밍 언어로 표현할 때에는 재귀함수로 구현하고 실행시킬 수 있다.
세 번째, 가능한 상태를 작은 수부터 큰 수로 만들어가는 방법
일단, a b와 상관없이 저장되는 값만 중요하고, 작은 수부터 나열하면 초기에
5, 7 이다. 가능한 두 수를 합해서 만들어지는 수는 12 이므로 12를 오른쪽에 추가한다.
5, 7, 12 이다. 12에 5를 더하고, 12에 7을 더하면 만들 수 있는 수 17, 19를 오른쪽에 추가한다.
5, 7, 12, 17 19 상태에서 17에 5, 7, 12를 각각 더해 오른쪽에 크기순으로 나열한다.
5, 7, 12, 17, 19, 22, 24, 29 상태에서 ... 19에....
... 이런 식으로 계속 해 가면 안 되는 수를 찾을 수 있다.

15. 다음 도형에서 작은 사각형은 모두 가로길이와 세로길이가 1인 정사각형이다. 이 도형의 한 점 A에서 출발하여 선분을 따라 움직이면서, 도형의 모든 선분을 지나 A로 다시 돌아오고자 한다. 같은 선분을 두 번 이상 지나도 된다. 지나는 거리의 최솟값은 얼마인가?

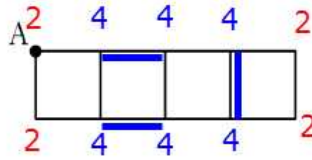


① 13 ② 14 ③ 15 ④ 16 ⑤ 17

** 출제의도?
그래프 이론? 오일러 한 붓 그리기?
** 의미?
형태가 간단하기 때문에 시행착오 방법으로도 가능하지만, 생각과 아이디어를 이용해 최솟값을 찾을 수 있다. 오일러 한 붓 그리기로 변형 응용?
** 해결예시?
간단하기 때문에 이렇게 저렇게 해보면서 시도해보는 시행착오(heuristic, 휴리스틱) 방법으로도 가능하지만, 발견한 방법이 최솟값임을 증명하기는 만만하지 않기 때문에, "더 적은 방법이 있을까?" 계속 고민하면서 시도해 보아야 한다.(예를 들어 8자 형태로 돌아서 가는 길과 오는 길을 최대한 겹치지 않도록 경로 선택?)
하지만, 오일러 한 붓 그리기와 유사한 형태이기 때문에, 그 원리를 이용하면 쉽고 빠르게 최솟값을 찾아나갈 수 있다. 오일러 한 붓 그리기는 간선으로 연결된 정점으로 표현할 수 있는 형태(그래프)에서 연결된 간선의 개수에 따라 한 붓 그리기가 가능한지 아닌지를 판단할 수 있는 것으로, 연결된 간선의 개수가 홀수인 점이 2개뿐인 경우 한 점에서 출발해 다른 한 점에서 끝내는 한 붓 그리기를 할 수 있고, 모든 간선의 개수가 짝수라면, 어디에서 시작하던지 다시 돌아오는 한 붓 그리기가 가능하다는 핵심 아이디어를 가지고 있다. 특히, 모든 점이 짝수인 경우에는 시작점에서 시작해서 다 돌고, 원래 시작점으로 돌아올 수 있다는 것에 대한 아이디어가 중요하다.
이 문제에 그런 한 붓 그리기 조건을 응용하면 보다 빠르게 정확한 값을 얻어낼 수 있게 된다.
아래 그림은 각 정점들에 연결된 간선의 개수를 표시한 것이다.



이제 한 붓 그리기가 가능한지 살펴보면? 불가능하다는 것을 알 수 있다. 이 상태에서 '한 붓 그리기가 가능하도록 만들어보자' 라는 생각을 한다면? '간선의 개수가 홀수인 점들을 적당히 더 연결하면 될 수 있지 않을까?' 라는 가능성을 생각해 볼 수 있다.
홀 수점들을 서로 연결하면 정점들의 간선의 개수가 짝수가 되면서, 한 분 그리기가 가능하기 때문에 A에서 시작해서 A로 돌아오는 어떤 경로가 반드시 있음을 알 수 있다. 그런데, 그렇게 연결했을 때 최단으로 만들어야 한다. 그림에서 보여지는 홀 수 점들이 6개가 있기 때문에 6개를 적당히 쌍으로 묶을 수 있다. 하지만, 아무렇게나 연결시키면 그게 최소일까? 아쉽게도 아니다. 그럼 어떻게 하면 좋을까? 선을 새로 연결한다는 것은 그냥 그렇게 두 점 사이를 움직일 수 있는 길을 다시 사용한다는 의미로 생각하고 접근하는 것이다.
이제 홀수 점들을 서로 연결시키면서 최소(합)이 되도록 만들 수 있는 방법을 생각해보자.
홀수 점들을 서로 연결해서 짝수 점들로 만드는데 가장 인접한 것들을 순서대로 연결해보면, 가능한 예시들 중 하나는 아래 그림과 같다.



모든 점들의 간선의 개수가 짝수이니 반드시 한 붓 그리기로 돌아올 수 있게 되었다! 이렇게 변형된 경로를 한 붓 그리기로 돌아서올 때, 필요한 길이는 전체막대의 개수+3개가 된다.

그래프가 더 복잡해지면 다른 전략도 사용할 수 있다고 추측되는데 이는 중고등부의 문제를 통해 설명한다.

16. C나 C++ 언어로 작성된 프로그램의 실행이 시작되는 함수는 다음 중 어느 것인가?

① start( ) ② main( ) ③ end( ) ④ up( ) ⑤ foo( )

** 출제의도?
C언어 기초
** 의미?
main 함수
** 해결예시?

17. 다음과 같이 정의된 배열 a는 모두 몇 개의 int형 변수로 구성되는가?

```
int a[3][4][5];
```

① 12개 ② 15개 ③ 20개 ④ 60개 ⑤ 240개

** 출제의도?
C언어 기초
** 의미?
다차원 배열의 선언
** 해결예시?
C언어에서 배열을 사용하기 위해 먼저 선언(정의)할 때, 필요한 개수만큼 선언한다. 여러 차원으로 선언되면 그 곱만큼 경우의 수(공간)를 만들 수 있다.

18. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, t;
for(i=0; i<=10; i++){
  if(i%7==0) t=i;
}
printf("%d\n", t);
```

① 7 ② 1 ③ 10 ④ 9 ⑤ 0

** 출제의도?
C언어 코딩의 이해(문법 암기가 아님)
** 의미?
0~10 범위의 정수 중에서 7로 나누어 나머지가 0인 경우 중, 최대 값 찾기
** 해결예시?

19. 실수 0.5의 값을 정확히 저장할 수 있는 변수의 자료형은 다음 중 어느 것인가?

① int ② long ③ double
④ short ⑤ char

** 출제의도?
C언어 기초
** 의미?
데이터형의 저장 범위/정확도
** 해결예시?
참고: 과거의 표준 C에서 32비트 정수 데이터형인 long 이 현재의 표준 C에서는 int로 되었다. 즉, int가 32비트(-2147483648 ~ +2147483647) 정수형으로 확대된 것이다.

20. 다음 중에서 C나 C++ 언어의 키워드(keyword)가 아닌 것은?

① float ② getchar ③ int
④ return ⑤ case

** 출제의도?
C언어 기초
** 의미?
의미가 이미(미리) 정해져있는 예약어가 아닌 것 구분하기
** 해결예시?
참고: 예약어는 변수/함수로 사용할 수 없으며, IDE에 따라 진한 색으로 잘 구분되도록 표시되기도 한다.

21. 다음 프로그램의 출력 결과는 무엇인가?

```
int a=20, b=10;
a=a+b;
b=a-b;
a=a-b;
printf("%d %d\n", a, b);
```

① 10 20 ② 20 20 ③ 10 10
④ 20 10 ⑤ 30 10

** 출제의도?
값 바꾸기의 다른 방법에 대한 이해
** 의미?
일반적으로 임시 변수를 이용해서 두 변수에 저장되어있는 값을 바꾸지만, 임시 변수 없이도 계산으로 두 변수의 값을 바꿀 수 있다는 의미한다.
** 해결예시?
참고 : 계산하는 (순차실행)순서가 매우 중요하다.

22. 다음 프로그램의 출력 결과는 무엇인가?

```
printf("Hello ");
printf("World!\n");
```

① Hello ② Hello World! ③ HelloWorld
④ HelloWorld!n ⑤ “Hello ”“World!”

** 출제의도?
C언어 기초
** 의미?
형식에 맞추어 출력하는 printf( ) 함수에 대한 이해. 특정한 동작/작업을 의미하는 이스케이프 문자 ‘\n’(new line)에 대한 이해
** 해결예시?

23. 다음 프로그램의 출력 결과는 무엇인가?

```
int a, b;
a=(16/3)*3;
b=16/(3*3);
printf("%d %d\n", a, b);
```

① 16 0 ② 16 2 ③ 15 1
④ 16 1 ⑤ 15 2

** 출제의도?

정수/정수 연산의 정확한 이해. 계산 순서에 대한 명확한 이해

** 의미?

정수의 계산과 실수의 계산 매카니즘이 다르기 때문에 C언어에서 정수/정수는 정수의 결과값을 가지게 되고, 그 형태는 자연스럽게 몫의 의미를 가진다. 일반적인 사칙연산과 동일한 계산 순서를 포함하기 때문에 괄호부터 먼저 계산되어 값으로 바뀐 후, 다음 순서의 계산이 순서대로 진행된다.

계산을 표현한 식(expression) 이라고 부르고, 그 식의 의미가 평가(evaluate)되어 계산된 결과 값(value)이 순서대로 하나씩 만들어지며 계산된다.

** 해결예시?

24. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[5]={1,-2,3,-2,4};
printf("%d\n", a[2]+a[3]-a[1]);
```

① 0 ② 1 ③ 2 ④ 3 ⑤ 4

** 출제의도?

배열의 참조에 대한 명확한 이해

** 의미?

배열의 선언과 동시에 초기값 넣기 방법의 이해. 배열의 값 참조를 통한 계산 이해. C언어의 배열 참조 번호는 0번부터 사용된다.

** 해결예시?

2번 위치에 있는 값에서 3번 위치에 있는 값을 더한 후, 1번 위치에 있는 값을 빼면 어떻게 될까?

25. 다음 프로그램의 출력 결과는 무엇인가?

```
int a=61, b=9, c=0;
while(a>=b){
  a=a-b;
  c=c+1;
}
printf("%d\n", c);
```

① 4 ② 5 ③ 6 ④ 7 ⑤ 8

** 출제의도?

코딩 이해

** 의미?

빼기 연산을 이용한 몫 계산?

** 해결예시?
크거나 같은 수에서 작은 수를 계속적으로 빼면서 카운팅 하고 있다. 결국 61/9의 몫의 의미와 같게 된다.

26. 다음 프로그램의 출력 결과는 무엇인가?

```
int a=2014, b=0;
while(a%2==0){
  a/=2;
  b++;
}
printf("%d\n", b);
```

① 0 ② 1 ③ 2 ④ 3 ⑤ 4

** 출제의도?
코딩 이해
** 의미?
단순 트레이싱?
** 해결예시?
짝수이면 반으로 나누고 카운팅을 반복한다.

27. 다음 프로그램의 출력 결과는 무엇인가?

```
int a=2014, b=380;
while(b!=0){
  int t;
  t=a%b;
  a=b;
  b=t;
}
printf("%d\n", a);
```

① 0 ② 1 ③ 38 ④ 114 ⑤ 2014

** 출제의도?
코딩의 이해
** 의미?
유클리드 호제법(수학자 유클리드가 만들어낸 서로 빼기 방법을 통한 최대 공약수 계산하기)의 구현
** 해결예시?
참고: 나머지 연산자는.. 빼기 횟수를 줄여준다. 빼기 연산으로도 유클리드 호제법을 구현

할 수 있다.

28. f(13, 20)의 값은 무엇인가?

```
int f(int a, int b){
  int i, c=0;
  for(i=a; i<=b; i++){
    f(i%2==0) c=c+i;
  }
  return c;
}
```

① 4　　② 33　　③ 68　　④ 70　　⑤ 132

** 출제의도?
  함수의 이해
** 의미?
  a이상 b 이하([a, b]) 의 짝수의 개수
** 해결예시?
  13 ~ 20 까지 짝수의 개수를 세어 보면 된다.

29. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, a=144, b=156, tot=0;
for(i=1; i<=100; i++){
  if(a%i==0 && b%i==0) tot=tot+i;
}
printf("%d\n", tot);
```

① 0　　② 1　　③ 12　　④ 28　　⑤ 36

** 출제의도?
  코드 이해
** 의미?
  두 수의 공약수의 합
** 해결예시?
  144, 156의 공약수 중 100보다 작은 값들의 합이다.

30. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, tot=0;
int a[10]={10,37,23,4,8,71,23,9,52,41};
```

```
for(i=0; i<10; i++){
  tot=tot+a[i];
  if(tot>=100) break;
}
printf("%d\n", tot);
```

① 82　　② 100　　③ 143　　④ 153　　⑤ 176

** 출제의도?
코드 이해
** 의미?
배열 누적 합
** 해결예시?
배열의 값을 순서대로 누적해가면서 누적합을 만들다가, 그 누적합이 100이상이 되면 중지하고 그 누적합을 출력한다.

31. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, b=0;
int a[11]={10,-2,3,4,8,6,-9,0,12,1,-3};
for(i=0; i<11; i++){
 if(i%3==1) b=b-a[i];
 else if(i%3==2) b=b+a[i];
}
printf("%d\n", b);
```

① 12　　② 18　　③ 24　　④ 27　　⑤ 30

** 출제의도?
코드이해
** 의미?
…
** 해결예시?
3으로 나눈 나머지가 1이면 합에서 빼고, 3으로 나눈 나머지가 2이면 합에 더하고, 3의 배수는 그냥 지나친다.

32. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, j, k, count=0;
int a[5]={4,5,3,1,2};
for(i=0; i<5; i++){
  for(j=4; j>i; j--){
    if(a[j-1]>a[j]){
```

```
        k=a[j-1];
        a[j-1]=a[j];
        a[j]=k;
        count++;
      }
   }
}
printf("%d\n", count);
```

① 5　　② 6　　③ 7　　④ 9　　⑤ 10

** 출제의도?

코드 이해

** 의미?

삽입 정렬의 다른 구현. 가장 오른쪽 데이터부터 왼쪽 데이터랑 비교해 계속 교환해 들어간다.

** 해결예시?

삽입 정렬시 데이터 교환 횟수를 출력하고 있다.

[4, 5, 3, 1, 2] 상태에서 시작해서가장 오른쪽 2부터 왼쪽 데이터와 비교해 간다. (3, 1)에서 교환이 발생한다.

[4, 5, 1, 3, 2] 상태에서 (5, 1), 교환이 발생한다.

[4, 1, 5, 3, 2] 상태에서 (4, 1), 교환이 발생한다.

[1, 4, 5, 3, 2] .. 상태가 되면서 한번 pass 진행

[1, 4, 5, 3, 2] 상태에서 (3, 2) 교환이 발생한다.

[1, 4, 5, 2, 3] 상태에서 (5, 2) 교환이 발생한다.

[1, 4, 2, 5, 3] 상태에서 (4, 2) 교환 발생이 발생한다.

계속 해가면 총 교환 횟수를 구할 수 있다.

33. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[4]={2,6,8,3};
int b[4]={1,0,3,2}, c[4]={2,1,3,0};
void f(int a[], int b[], int n){
  int i;
  for(i=0; i<n; i++) a[i]=a[b[i]];
}
int main(){
 int i;
 f(a, b, 4);
 f(a, c, 4);
 for(i=0; i<4; i++) printf("%d ", a[i]);
 printf("\n");
```

```
 return 0;
}
```

① 2 6 8 3　　② 3 2 8 6　　③ 3 6 3 3
④ 6 2 3 8　　⑤ 6 6 3 3

** 출제의도?
코드 이해, 함수의 이해
** 의미?
함수의 이해, 배열의 이중참조에 대한 이해
** 해결예시?
f(a, b, n) 함수는 각각 n개의 원소를 가지는 a, b 배열을 받아서, b배열에 저장되어있는 값을 참조해서 a배열의 값을 바꾸는 함수이다.
배열의 이중참조(더블 인덱싱)를 헷갈리지 않도록 주의해야 한다.

34. 다음 프로그램의 출력 결과는 무엇인가?

```
int a=1, b=3, c=0, n=3;
while(n>=0){
 c=b+a;
 a=b;
 b=c;
 n=n-1;
}
printf("%d\n", c);
```

① 4　　② 7　　③ 11　　④ 16　　⑤ 18

** 출제의도?
코드 이해
** 의미?
카운팅 코드? n이 3이면 4번 반복되는 코드? 처음에 1, 3으로 시작하는 피보나치 수 만들기?
** 해결예시?
1, 3, 4
1, 3, 4, 7
1, 3, 4, 7, 11
1, 3, 4, 7, 11, 18

35. 다음 프로그램의 출력 결과는 무엇인가?

```
char s[7]="banana";
int i, j, c=0;
for(i=0; i<6; i++){
  for(j=0; j<=i; j++) if(s[i]!=s[j]) c++;
}
printf("%d\n", c);
```

① 11 ② 12 ③ 13 ④ 14 ⑤ 15

** 출제의도?
코드 이해
** 의미?
어떤 문자 이후로 그 문자와 다른 문자 개수 카운팅
** 해결예시?
b와 다른 것 5개, a와 다른 것 2개, n과 다른 것 2개이다.

36. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, b=0, c=0;
int a[8]={-3,8,7,-6,1,9,3,-5};
for(i=0; i<8; i++){
  if(i%2==0 && b>a[i]) b=a[i];
  if(i%2==1 && c<a[i]) c=a[i];
}
printf("%d\n", c-b);
```

① 3 ② 6 ③ 9 ④ 12 ⑤ 15

** 출제의도?
코드 이해
** 의미?
짝수 번째 수들 중 가장 작은 수 b, 홀 수 번째 수들 중 가장 큰 수 c.
** 해결예시?
c-b ?

37. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, tot=0, s=1;
for(i=1; i<=10; i++){
  if(i%2==0){
    tot=tot+s*i;
```

```
    s=-s;
  }
}
printf("%d\n", tot);
```

① 6 ② 30 ③ 5 ④ -5 ⑤ 10

** 출제의도?
코드 이해.
** 의미?
1~10 까지 2의 배수는 더하고, 4의 배수는 뺀다.
** 해결예시?

38. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, j, a[40];
for(i=0; i<40; i++) a[i]=0;
for(i=1; i<40; i++){
  for(j=i; j<40; j+=i) a[j]+=i;
}
printf("%d\n", a[24]);
```

① 40 ② 54 ③ 56 ④ 58 ⑤ 60

** 출제의도?
코드 이해
** 의미?
약수의 합?
** 해결예시?
24의 약수는 1, 2, 3, 4, 6, 8, 12, 24 이다.

39. 다음 보기 중 결과값이 1인 것은 무엇인가?

```
int f(int a){
  int i, r;
  r=0;
  for(i=1; i<=a; i++){
    if(a%i==0) r=1-r;
  }
  return r;
}
```

① f(121) ② f(123) ③ f(127)
④ f(128) ⑤ f(255)

** 출제의도?
  코드 이해, 함수의 이해
** 의미?
  완전제곱수 찾기?
** 해결예시?
  어떤 수의 약수의 개수가 홀수 개이어야만 r이 1이 되므로, 완전 제곱수 이어야만 한다.

40. 다음 프로그램의 출력 결과는 무엇인가?

```
char enc[12]="mississippi";
int i, j;
for(i=0; i<3; i++){
  for(j=i; j<11; j+=3) printf("%c", enc[j]);
}
printf("\n");
```

① iiiissssppm ② ippississim ③ mississippi
④ msspisspiii ⑤ msspiiiissp

** 출제의도?
  코드 이해
** 의미?
  3개의 부분으로 쪼갠 후, 각 부분별로 하나씩 출력한다.
** 해결예시?
  mis sis sip pi
  맨 앞에 문자 하나씩 출력하면 mssp
  두 번째 문자 하나씩 출력하면 iiii
  세 번째 문자 하나씩 출력하면 ssp

41. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[9]={5,7,6,4,2,1,3,9,8};
int rank_a[9], rank_b[9];
int i, j;
for(i=0; i<9; i++){
  rank_a[i]=rank_b[i]=1;
  for(j=0; j<9; j++){
    if(a[j]>a[i]) rank_a[i]++;
    if(a[j]>a[i] && j<i) rank_b[i]++;
  }
}
```

```
for(i=0; i<9; i++) printf("%d ", rank_a[i]);
printf("\n");
for(i=0; i<9; i++) printf("%d ", rank_b[i]);
printf("\n");
```

① 5 3 4 6 8 9 7 1 2
1 1 2 4 5 6 5 1 2

② 5 7 6 4 2 1 3 9 8
1 2 2 1 1 1 3 8 8

③ 5 7 6 4 2 1 3 9 8
0 0 1 3 4 5 4 0 1

④ 4 6 5 3 1 0 2 8 7
0 1 1 0 0 0 2 7 7

⑤ 5 3 4 6 8 9 7 1 2
8 8 7 5 4 3 4 8 7

** 출제의도?
코드 이해.
** 의미?
랭킹 계산 예시. 배열을 활용한 계산. 이해.
** 해결예시?
a[ ] 는 점수
rank_a[ ] 는 등수
rank_b[ ] 는 자기(a[i]) 보다 큰 값이 왼쪽에 몇 개 있는가?

42. 다음 프로그램의 출력 결과는 무엇인가?

```
void f(int mat[2][2]){
  int i, j;
  int tmp[2][2];
  for(i=0; i<2; i++){
    for(j=0; j<2; j++) tmp[i][j]=mat[1-i][j];
  }
  for(i=0; i<2; i++){
    for(j=0; j<2; j++) mat[i][j]=tmp[i][j];
  }
}
int main(){
  int mat[2][2]={{1,2}, {2,1}};
  int i, j;
  for(i=0; i<999; i++) f(mat);
  for(i=0; i<2; i++){
    for(j=0; j<2; j++) printf("%d ", mat[i][j]);
    printf("\n");
  }
```

```
  return 0;
}
```

① 2 1
1 2 ② 1998 999
999 1998 ③ 1 2
1 2

④ 1 2
2 1 ⑤ 2 1
3 2

** 출제의도?

코드 이해. 함수의 이해. 행렬의 계산.

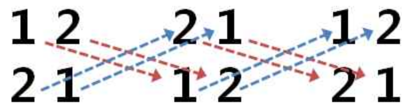
** 의미?

함수에 배열 전달하기

** 해결예시?

f( ) 함수는 전달된 배열에 적당한 작업을 한 후, 원래의 배열에 그 결과를 복사해 넣는 작업을 한다.

f( ) 함수를 몇 번 실행하면? 어떻게 되는지 관찰해 본다.



1번 실행

2번 실행

999번 실행하면 어떻게 될까?

43. 다음과 같은 결과를 얻는 프로그램을 작성하고 싶다. 빈 칸(a)와 (b)에 알맞은 구문은?

```
 1  2  3  4  5  6
12 11 10  9  8  7
13 14 15 16 17 18
24 23 22 21 20 19
```

```
int a[4][6];
int y, x;
for(y=0; y<4; y++){
  for(x=0; x<6; x++){
    if(y%2==0) a[y][x]= (a) ;
    else a[y][x]= (b) ;
  }
}
for(y=0; y<4; y++){
  for(x=0; x<6; x++) printf("%2d ", a[y][x]);
```

```
  printf("\n");
}
```

① (a) y * 6 + 6 - x (b) y * 6 + x + 1 ② (a) y * 6 + x (b) y * 6 + x − 1 ③ (a) y * 4 + x + 1 (b) y * 4 + 6 − x

④ (a) y * 6 + x + 1 (b) y * 6 + 6 − x ⑤ (a) y * 6 + 1 + 2 + 3 (b) y * 6 + 6 + 5 + 4

** 출제의도?
  코드 이해, 패턴 파악, 계산식 만들기
** 의미?
  2차원 배열 채우기, 지그재그 채우기.
** 해결예시?
  첫 줄부터 0, 1, 2, 3 줄, 왼쪽 열부터 0, 1, 2, 3, 4, 5 열 이라고 생각하면, 짝수 줄은 왼쪽에서 오른쪽으로 채워나가고, 홀수 줄은 오른쪽에서 왼쪽으로 채워나가는 지그재그 배열 채우기이다.

```
 1  2  3  4  5  6
12 11 10  9  8  7
13 14 15 16 17 18
24 23 22 21 20 19
```

 출력결과를 잘 살펴보면 y줄 x열 의 위치를 채우는데.. 짝수 줄인 경우 6*y+x+1 의 값이 채워짐을 생각할 수 있다. 예를 들어 14는 2줄, 1열인데 그 값은 14이다.((6*2) + 1 + 1) 중요한 것은 이러한 계산식을 스스로 생각해 낼 수 있어야 한다는 것이다.

44. f(0, 7)의 값은 무엇인가?

```
int c[8]={1,3,7,2,4,1,5,3};
int f(int a, int b){
  int mid;
  if(a==b) return c[a];
  else{
    mid=(a+b)/2;
    return f(a, mid)−f(mid + 1, b);
  }
}
```

① -8 ② -5 ③ 0 ④ 5 ⑤ 6

** 출제의도?
  재귀 함수의 이해
** 의미?
  다중 재귀의 이해
** 해결예시?

두 구역으로 나누어 왼쪽 값에서 오른쪽 값을 빼는 형태. 구역을 계속 나누어간 후 재귀적으로 계산해 올라오면 된다.

45. 다음 보기 중에서 가장 큰 값은 무엇인가?

```
int f(int n){
  if(n==0) return 0;
  return (n%10)+f(n/10)+f(n/100);
}
```

① f(14)　② f(77)　③ f(96)
④ f(151)　⑤ f(512)

** 출제의도?
재귀 함수의 이해?
** 의미?
3자리 10진수에 대해서, 각 자리 수의 합
** 해결예시?
각 자리 숫자의 합을 계산해 보면 된다.

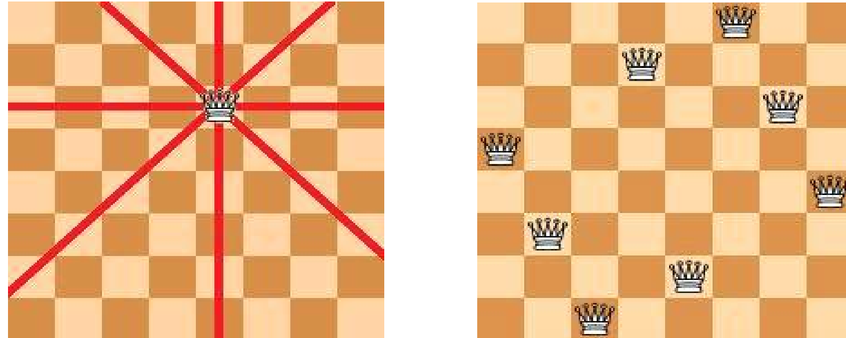
46. 다음 프로그램의 출력 결과는 무엇인가?

```
void f(char *str, char *dest){
  int i, j;
  for(i=0, j=0; str[i]!='\0'; i++){
    if(str[i]!=str[i+1]){
      dest[j]=str[i];
      j++;
    }
  }
  dest[j]='\0';
}
int main(){
  char in[20]="aaabccceef", out[20];
  f(in, out);
  printf("%s\n", out);
  return 0;
}
```

① aaabccceef　② abcdef　③ abcef
④ aabccef　⑤ ace

** 출제의도?
코드 이해? 함수에 배열 포인터 전달하기?
** 의미?
반복 문자 제거하기
** 해결예시?
aaabcccceef에서 반복되는 문자를 1개씩만 남겨둔다.

[47-48] 체스 게임에서 여왕말은 한 턴에 상하좌우 그리고 두 대각선 방향으로 거리에 상관없이 공격 할 수 있다. 이 때, 크기가 n×n인 격자에 n개의 여왕말들을 서로 공격할 수 없도록 배치하는 문제를 n-queen문제라고 한다.



여왕말의 공격범위 n=8 일 때 해법

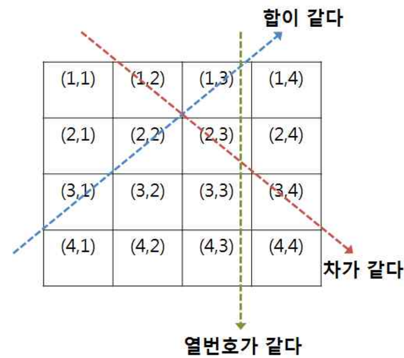
위의 오른쪽 그림은 n=8 일 때의 n-queen문제의 해법 중 하나이다. 다음은 n=4 일 때 n-queen문제의 해법의 개수를 세는 코드이다.

```
int f(int a, int b, int c, int d){
  int arr[4]={a,b,c,d};
  int i, j;
  for(i=0; i<4; i++){
    for(j=0; j<4; j++){
      if(i==j) continue;
      if(arr[i]==arr[j] || (a) || (b)) return 0;
    }
  }
  return 1;
}
int main(){
  int i, j, u, v, count=0;
  for(i=0; i<4; i++){
    for(j=0; j<4; j++){
      for(u=0; u<4; u++){
       for(v=0; v<4; v++) count+=f(i, j, u, v);
      }
    }
  }
  printf("%d\n", count);
  return 0;
}
```

47. 올바른 결과를 얻기 위해 빈 칸 (a)와 (b)에 알맞은 구문은?

① (a) arr[i] + arr[j] == i + j　　(b) arr[i] - arr[j] == i - j
② (a) arr[i] + arr[j] + i == 1　　(b) arr[i] + arr[j] + j == 1
③ (a) arr[i] - arr[j] == i - j　　(b) arr[i] + arr[j] == i + j
④ (a) arr[i] + i == arr[j] + j　　(b) arr[i] - i == arr[j] - j
⑤ (a) arr[i] ^ arr[j] == 0　　(b) arr[i] ^ 0 == arr[j]

** 출제의도?
코드 이해
** 의미?
n-queen 예시 해법. 단, 아주 작은 경우에 대한 예시 해법
4개의 줄에서 4개의 열 번호 조합을 모두 만들어서, 그 모든 경우에 대해 가능한지 판단/카운팅한다.
** 해결예시?
4개의 좌표를 (x, y)라고 생각하고 그림을 그리고 관찰을 해보면, 아래와 같은 특성을 찾아낼 수 있다.

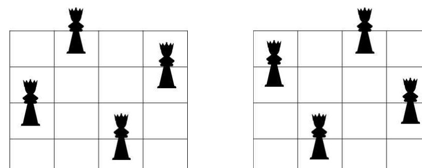


이러한 특성을 퀸을 놓을 수 있는지 검사에 사용하면 된다. 이러한 특성을 검사할 수 있는 식을 찾는다.

48. 코드가 올바르게 작성되었을 때, 프로그램의 출력 결과는 무엇인가?

① 0　② 1　③ 2　④ 3　⑤ 4

** 출제의도?
실제 작은 데이터에 대한 문제 이해 여부 파악?
** 의미?
문제/상황을 정확히 이해하고 간단한 상황에 대해서 해결할 수 있는지?
** 해결예시?
가장 윗줄에서 하나씩 채워 보면 2가지만 가능함을 찾을 수 있다.



[49-50] 다음과 같은 문제를 해결하기 위해 프로그램을 작성하였다. 물음에 답하시오.

올림픽은 참가에 의의가 있기에 공식적으로는 국가간 순위를 정하지 않는다. 그러나 많은 사람들이 자신의 국가가 얼마나 잘 하는지에 관심이 많기 때문에 비공식적으로는 국가 간 순위를 정하고 있다. 두 나라가 각각 얻은 금, 은, 동메달 수가 주어지면 보통 다음 규칙을 따라 어느 나라가 더 잘했는지 결정한다.

1. 금메달 수가 더 많은 나라 2. 금메달 수가 같으면, 은메달 수가 더 많은 나라 3. 금, 은메달 수가 모두 같으면, 동메달 수가 더 많은 나라

각 국가는 1부터 사이의 정수로 표현된다. 한 국가의 등수는 (자신보다 더 잘한 나라 수) + 1로 정의된다. 만약 두 나라가 금, 은, 동메달 수가 모두 같다면 두 나라의 등수는 같다. 예를 들어, 1번 국가가 금메달 1개, 은메달 1개를 얻었고, 2번 국가와 3번 국가가 모두 은메달 1개를 얻었으며, 4번 국가는 메달을 얻지 못하였다면, 1번 국가가 1등, 2번 국가와 3번 국가가 공동 2등, 4번 국가가 4등이 된다. 이 경우 3등은 없다.

각 국가의 금, 은, 동메달 정보를 입력받아서, 어느 국가가 몇 등을 했는지 알려주는 프로그램을 작성하시오.

입력 형식

입력파일의 이름은 input.txt로 한다. 입력의 첫 줄은 국가의 수 $N(1 \le N \le 1,000)$과 등수를 알고 싶은 국가 $K(1 \le K \le N)$가 빈칸을 사이에 두고 주어진다. 각 국가는 1부터 N사이의 정수로 표현된다. 이후 N개의 각 줄에는 차례대로 각 국가를 나타내는 정수와 이 국가가 얻은 금, 은, 동메달의 수가 빈칸을 사이에 두고 주어진다. 전체 메달 수의 총합은 1,000,000 이하이다.

출력 형식

출력파일의 이름은 output.txt 이다. 출력은 단 한 줄이며, 입력받은 국가 K의 등수를 하나의 정수로 출력한다. 등수는 반드시 문제에서 정의된 방식을 따라야 한다.

입력과 출력의 예1

입력

```
4 3
1 1 2 0
2 0 1 0
3 0 1 0
4 0 0 1
```

출력
2

입력과 출력의 예2

입력
4 2
1 3 0 0
3 0 0 2
4 0 2 0
2 0 2 0

출력
2

프로그램

```
#include <stdio.h>
int medal[1001][4];
int main(){
  int a, I, n, k, rank=1;
  FILE *fi, *fo;
  fi = fopen("input.txt", "r");
  fscanf(fi, "%d %d", &n, &k);

  for(i=0; i<n; i++){
    fscanf(fi, "%d", &a);
    fscanf(fi, "%d %d %d", &medal[a][1], &medal[a][2], &medal[a][3]);
  }
  fclose(fi);
  for(i=1; i<=n; i++){
    if(medal[i][1] > medal[k][1] ||
    (medal[i][1] (a) medal[k][1] && medal[i][2] (b) medal[k][2]) ||
    (medal[i][1] (a) medal[k][1] && medal[i][2] (a) medal[k][2] &&
     medal[i][3] (b) medal[k][3])) rank++;
 }

 fo=fopen("output.txt", "w");
 fprintf(fo, "%d \n", rank);
```

```
 fclose(fo);
 return 0;
}
```

49. 다음 중 (a) 와 (b) 에 들어갈 내용으로 알맞은 것은?

① (a) == (b) >　　② (a) < (b) >　　③ (a) != (b) <

④ (a) > (b) ==　　⑤ (a) == (b) <

** 출제의도?

코드 이해

** 의미?

문제/상황에 적합한 계산식 채우기

** 해결예시?

50. 입력 파일 input.txt가 다음과 같이 주어졌을 때, 위 프로그램의 출력은 무엇인가?

```
8 4
1 4 4 4
5 4 4 5
2 4 5 3
4 4 3 10
6 3 6 9
3 5 8 0
7 3 10 0
8 4 3 7
```

① 3　　② 4　　③ 5　　④ 6　　⑤ 7

** 출제의도?

문제/상황 이해

** 의미?

코드화된 데이터를 보고 문제/상황에 맞는 답 계산하기

** 해결예시?

2014 시도예선 중고등부 문제

1. 5개의 수 30, x, y, z, 50에 대하여, x, y, z는 각각 왼쪽에 있는 수와 오른쪽에 있는 수의 평균값이다. x+y+z의 값은?

   ① 120 ② 125 ③ 127 ④ 135 ⑤ 140

** 출제의도?

계산/논리식 만들기. 수학적 통찰. 값 구하기.

** 의미?

주어진 문제/상황에서 관계식을 만들고 미지수 계산

** 해결예시?

문제에서 평균값에 대해서 각 x, y, z에 대해서 계산식으로 표현하면, y = (x+z)/2, x=(30+y)/2, z=(y+50)/2 와 같다. 미지수가 3개이고, 서로 다른 식이 3개이므로 답을 계산해 낼 수 있다.

수학적인 통찰로서 주어진 상황에서 적당히 y의 값이 30과 50의 중간 값 이라고 생각하면, y=40, x=35, z=45가 될 수 있고 문제/상황에 대한 답으로 만족한다.

2. 72014의 마지막 자리의 숫자(1의 자릿수)는 얼마인가?

   ① 1 ② 3 ③ 5 ④ 7 ⑤ 9

** 출제의도?

패턴분석?

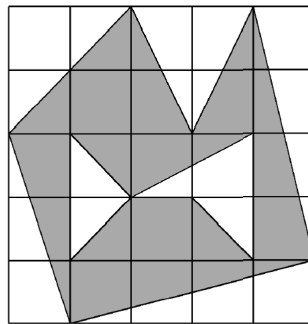
** 의미?

어떤 값의 거듭제곱에 의해 만들어지는 1의 자리숫자는 일정한 패턴/간격을 가지고 반복되는 경향이 있다.

** 해결예시?

7을 계속 거듭제곱하면서 만들어지는 1의 자리숫자를 살펴보고 그러한 패턴이 나타나는지 확인해 본다. 패턴/간격이 발견되면 그 패턴을 이용해 1의 자리숫자를 계산해 내면된다.

3. 아래 그림에서 가장 작은 정사각형은 모두 변의 길이가 1인 정사각형들이다. 색칠된 영역의 면적은 얼마인가?



   ① 11 ② 11.8 ③ 12
   ④ 12.5 ⑤ 13

** 출제의도?
빠른 계산? 패턴 분석?
** 의미?
여러 가지 모양/패턴을 구성할 수 있는 단위 패턴/길이를 이용해 빠르고 정확하게 계산
** 해결예시?
높이가 1인 밑변이 다른 삼각형으로 그림을 다시 나눈 후, 그 개수를 이용해 빠르고 정확하게 계산할 수 있다. 그 삼각형들의 밑변의 길이를 모두 합한 다음에 1/2를 곱하면 회색영역의 넓이가 된다.

4. 재석과 명수는 일직선 도로에서 함께 운동을 했다. 재석은 이 도로의 반을 시속 12km로 달리고, 나머지 반은 시속 4km로 달린 후, 시간을 재었더니 A 시간이 걸렸다. 명수는 도로의 처음 3분의 2는 시속 15km로 달리고 나머지는 시속 5km로 달린 후, 시간을 재었더니 B 시간이 걸렸다. A÷B의 값은?

① 5/4　② 4/3　③ 1　④ 7/5　⑤ 3/2

** 출제의도?
문제/상황 분석? 계산식 만들기?
** 의미?
주어진 문제/상황에서 계산식을 구성해 목표 답을 구해낸다.
** 해결예시?
거리/속력=시간이다. 일직선 도로의 길이를 L 이라고 하면,
재석이 달린 시간 A는 (L*1/2)/12+(L*1/2)/4=A로 표현할 수 있고,
명수가 달린 시간 B는 (L*2/3)/15+(L*1/3)/5=B 로 표현할 수 있다.
A/B 값을 구하면 된다.

5. A, B, C는 서로 다른 한자리 자연수들이다. 다음의 덧셈식이 성립한다면 A+B+C의 값은?

$$\begin{array}{r} A\ A\ A \\ B\ B\ B \\ +\ \ C\ C\ C \\ \hline C\ B\ B\ A \end{array}$$

① 11　② 13　③ 16　④ 18　⑤ 19

* 출제의도?
문제/상황 분석. 논리적 사고.
** 의미?
복면산 계산. 목표값을 찾기 위한 논리적 사고. 식의 변형.
** 해결예시?
주어진 문제/상황에서 고려해야할 것과 가능한 경우의 수가 꽤 많다. 물론, 논리적으로

가능한 경우들에 대해서 생각해 볼 수도 있지만 풀어야할 문제가 매우 많기 때문에 주어진 예시 답안을 이용해 목표값 A+B+C를 찾아 볼 수도 있다. 목표가 A, B, C를 각각 구할 필요는 없고 A+B+C 의 값만 구하면 되기 때문에 생각을 살짝 바꿔 주어진 계산식을 변형해서 생각해도 된다.

주어진 복면산 계산식에서 각 100, 10, 1의 자리 숫자들의 합을 보니 A+B+C이다. 이 관계를 생각하면 주어진 선택 답안들을 넣어 계산해보고 답인지 아닌지 찾아낼 수 있다. 이는 마치 “답이냐? 아니냐?”와 같은 형태로 해결해 보는 것으로, 이렇게 Yes/No의 답을 구하는 문제들을 결정 문제라고 한다.

주어진 문제는 가능한 A, B, C의 합을 찾는 것이지만, 주어진 예시 답안들로 계산해보고 거꾸로 된다?/안된다?의 결정문제로도 해결해 볼 수 있는 것이다.

잘 생각해보면, 예를 들어 11이 답이라고 하자. 그러면 주어진 복면산 식은

```
    11
   11
  11
+------
```

1221 로 변환한 것과 같다! 세로로 합한 값을 10배, 100배해서 더한 것이다. 이렇게 선택 답안들을 넣어보고 문제에서 주어진 조건들을 만족하는지 살펴보면 빠르게 답을 얻어낼 수 있다.

6. 다음 보기에는 각각 4행, 4열로 1부터 16까지의 수들이 표에 나열되어 있다. 5가지 보기 중에 4개는 서로 간에 여러 번의 행의 교환 혹은 열의 교환만으로 만들 수 있지만, 남은 하나는 그것이 불가능하다. 행 혹은 열의 교환만으로 만들 수 없는 보기는 무엇인가? 행의 교환이란 임의의 두 행 전체를 서로 바꾸는 것을 말한다. 마찬가지로 열의 교환은 임의의 두 열 전체를 서로 바꾸는 것을 뜻한다.

①

9	12	11	10
1	4	3	2
5	8	7	6
13	16	15	14

②

15	13	16	14
3	1	4	2
11	9	12	10
7	5	8	6

③

8	6	7	5
16	14	9	13
12	10	11	15
4	2	3	1

④

11	12	9	10
3	4	1	2
15	16	13	14
7	8	5	6

⑤

5	7	8	6
1	3	4	2
9	11	12	10
13	15	16	14

** 출제의도?

문제/상황 이해?

** 의미?

행끼리 바꾸기. 열끼리 바꾸기.

** 해결예시?

언뜻 복잡해 보이지만, 행끼리/열끼리 서로 자리를 바꿀 수 있는데.. 숫자가 커지는 순서는 행/열끼리 모두 같아야 한다. 아닌 것이 있다. 행끼리/열끼리 서로 바꾸어서 만들 수 없는 패턴이다.

7. 이차원 평면의 원점 (0, 0)에서 시작해서 한 번에 왼쪽, 오른쪽, 위쪽, 아래쪽 중 한 방향을 마음대로 정해 그 방향으로 거리 1만큼 이동하려고 한다. (0, 0)에서 한 번의 이동으로 갈 수 있는 곳은 (0, 1), (1, 0), (0, -1), (-1, 0)으로 모두 네 점이다. 그렇다면 정확히 10번 이동해서 도달할 수 있는 서로 다른 점의 개수는 몇 개인가?

① 100 ② 121 ③ 135 ④ 185 ⑤ 221

** 출제의도?

문제/상황 분석 및 이해? 논리적 사고? 아이디어?

** 의미?

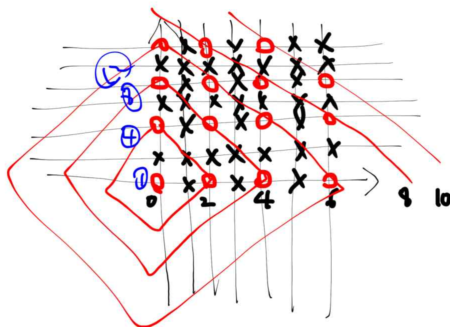
가능한 경우가 매우 많은 것처럼 보이지만, 간단한 아이디어와 분석으로 쉽게 해결할 수 있다.

** 해결예시?

2차원 형태로 주어지는 문제/상황이지만, 보다 쉬운 이해를 위해 1차원 직선에서 움직인다고 가정한 후 관찰해보고 2차원으로 확장 시킬 수 있다.

예를 들어 1차원 직선상에 정수 좌표들이 있다고 하자. 그 때 0번 좌표에서 시작해서 10번 이동해서 움직일 수 있는 좌표들은? 0 2 4 6 8 10 이다. 왜냐하면, 어떤 점에서 한번 오른쪽으로 이동했다가 다시 이전 좌표로 가면 2개의 이동이 짝수로 소모되기 때문에 절대로 홀 수 점에서 끝낼 수 없다.

이제, 같은 아이디어를 2차원으로 확장 시킨 후 그림을 그려서 문제를 해결할 수 있다.



점의 개수가 1, 4, 8, 12, ... 늘어나고 있다.

8. 동전이 10개씩 들어 있는 자루가 10개 있다. 이 중에 9개의 자루에는 모두 정상인 동전이 들어 있고, 나머지 하나의 자루에 들어 있는 동전은 모두 가짜이다. 정상인 동전 하나의 무게는 10g이고, 가짜 동전 하나의 무게는 9g이다. 무게를 그램(g) 단위로 잴 수 있는 저울을 최대 k번 이용하여 항상 가짜 동전이 들어 있는 자루를 찾을 수 있어야 한다. 가능한 가장 작은 k는 얼마인가?

① 1 ② 2 ③ 3 ④ 5 ⑤ 7

** 출제의도?

퍼즐 해결 아이디어.

** 의미?

많이 알려져 있는 문제. 양팔 저울이 아니고 무게를 잴 수 있는 저울이 주어짐. 해싱과 연관

** 해결예시?

순서대로 자루를 세워두고 1번~9번 자루라고 이름을 붙여두고. 각 자루에서 0개, 1개, 2개, ... (10-1)개를 꺼내 모은 후 무게를 1번 재면 된다. 만약 모든 자루에 들어있는 동전들이 모두 정상 동전이라면? 45개의 동전이 10g이므로 490g으로 나와야 한다. 그런데 무게를 잰 결과가 485g라면? 5g만큼 차이가 나는 것이므로 첫 번째 자루부터 1개, 2개, 3개, ... (10)개를 빼는 것과 차이가 없을까?

9. 속이 보이지 않는 주머니 안에 검은색 공이 20개, 흰색 공이 16개 있다. 여기서 두 개의 공을 보지 않고 꺼낸 다음, 두 공의 색이 같으면 새로운 검은색 공을 주머니에 넣고, 두 공의 색이 다르면 새로운 흰색 공을 넣는다. 이러한 과정을 주머니에 공이 한 개만 남을 때까지 반복하여, 남은 공의 색을 A라 하자. 초기에 검은색 공이 20개, 흰색 공이 15개일 때에 대해서 위의 과정을 똑같이 반복했을 때에 마지막에 남은 공의 색을 B라 하자. A와 B는 각각 무슨 색인지 알 수 있을까?

① A=검은색, B=검은색 ② A=검은색, B=흰색

③ A=흰색, B=검은색 ④ A=흰색, B는 검은색 또는 흰색 모두 가능

⑤ A, B 둘다 검은색 또는 흰색 모두 가능

** 출제의도?

상태 변화. 오토마타. 논리.

** 의미?

어떤 초기 상태에서 바뀔 수 있는 여러 가지 상태에서 다음 상태로의 변화에 대한 이해

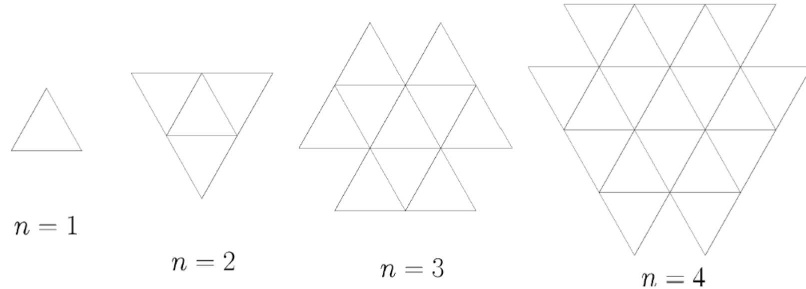
** 해결예시?

어떤 초기 상태를 (0(검은 공의 개수), 0(흰 공의 개수))이라고 하면 (0, 0) 상태에서 꺼낸 두 개의 공이 모두 검정이라면 (-1, 0) 상태로 바뀐다. 두 개의 공이 모두 흰색이라면 (+1, -2) 상태로 바뀐다. 그럼 검정 공 1개, 흰 공 1개라면 (-1, 0) 상태로 바뀐다. 결국 어떤 상태에서 가능한 변화 상태는 2가지뿐이다.

이러한 관계를 트리 구조로 표현할 수도 있다. 하지만 트리 구조로 (20, 16) 상태나 (20, 15) 상태로 시작해 아래로 내려 그려 내려가면 그려야 할 경우가 너무 많다. 여기서 논리적으로 생각해 볼 수 있다. 초기에 시작하는 흰색공의 개수가 짝수이면 무조건 검정 공이 남게 된다. 흰색 공의 개수가 홀수이면? 흰색이 남을 수밖에 없다. 왜냐하면 검정 공만 -1씩으로 줄어들기 때문이다. 이 논리가 잘 이해되지 않는다면 충분한 시간을 두고 생각해 보는 것이 좋다. 특히 가능한 변화 상태가 (-1, 0), (+1, -2) 뿐이라는 것을 잘 보고 생각하면 된다. 목표는 둘 중 하나의 공만 남을 때까지 변화시키는 것이다.

10. 어떤 알고리즘은 아래의 그림에서 보이는 바와 같이 n=1일 때에 작은 정삼각형에서 시작하여, n이 증가할 때마다, 바깥쪽 변에 작은 삼각형을

붙여나가는 방식으로 삼각형의 개수를 늘려간다고 한다. 예를 들어, n=4일 때에는 가장 작은 삼각형은 19개이다. n=11일 때 가장 작은 삼각형의 개수는 몇 개가 될까?



① 163 ② 164 ③ 165 ④ 166 ⑤ 199

** 출제의도?

패턴 분석. 재귀적 문제해결.

** 의미?

이전 상태와의 변환/증감관계 찾기. 재귀적 문제 해결의 기초.

** 해결예시?

이전 상태에서 늘어나는 조그만 삼각형의 개수를 작성해 보고 관계성을 파악한 후, 그 결과를 이용해 답을 찾는다.

n=1에서 n=2로 갈 때, 3개의 삼각형이 늘었다.

n=2에서 n=3로 갈 때, 6개의 삼각형이 늘었다.

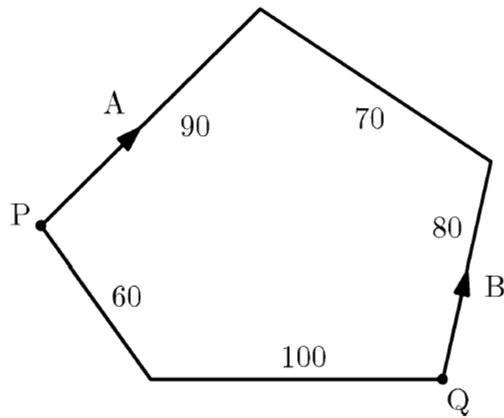
n=3에서 n=4로 갈 때, 9개의 삼각형이 늘었다.

한 번 더 직접 그려보면.

n=4에서 n=5로 갈 때, 12개의 삼각형이 늘어난다. 관계를 정리해보면 (n-1)*3 개씩 삼각형이 증가함을 예상할 수 있다.

증가되는 삼각형의 개수는 1(n=1), +3(n=2), +6, +9, +12, +15, ... , 30 (n=11) 처음 1개에서 늘어나는 개수들을 모두 더해 가면 n=11일 때의 전체 삼각형 개수를 계산할 수 있다.

11. 다음 그림과 같은 5각형 모양의 산책로를 걷는 두 사람 A, B가 있다. A는 점 P에서 출발하여 시계 방향으로 1분에 80m의 속도로, B는 점 Q에서 시계반대 방향으로 1분에 120m의 속도로 산책로를 걷고 있다. A와 B가 각각 점 P, Q에서 동시에 출발하여 A가 산책로를 100바퀴를 돌았을 때, A와 B는 몇 번 만나는가? 꼭짓점을 지날 때에서도 항상 같은 속도를 유지한다. 변 옆의 수는 변의 길이(m)를 나타낸다.



① 235　　② 240　　③ 245　　④ 250　　⑤ 255

** 출제의도?

문제/상황 분석. 패턴 찾기. 문제 단순화/변환.

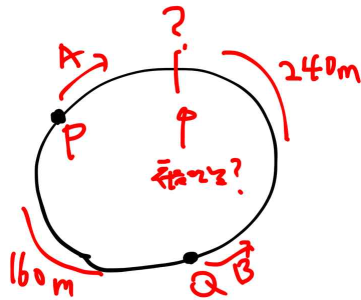
** 의미?

이벤트가 발생하는 주기 찾기

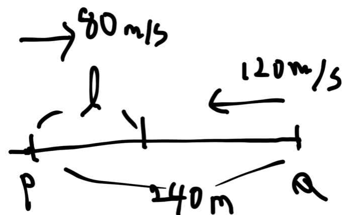
** 해결예시?

문제/상황을 명확히 이해하고 단순화하는 방법으로 답을 찾을 수 있다. 직선 형태의 다각형으로 표현되어있지만, 답을 찾기 위해서 다각형의 의미는 필요가 없다. 그리고 일단 한번 마주친 후에는 마치 원 트랙을 서로 반대로 도는 것과 같은 형태가 된다.

먼저 처음 만나는 위치를 계산하고 나서, 얼마마다 한 번씩 마주치게 되는지 주기를 파악하면, A가 100바퀴를 돌았을 때 몇 번 마주치는지를 간단히 계산해 낼 수 있게 된다.

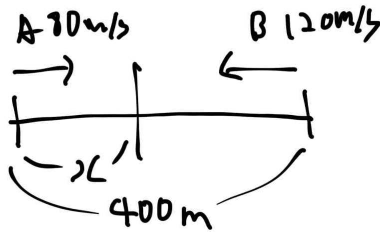


단순화 시킨 그림. 처음 마주치는 위치는 어디일까?



처음 만나는 위치를 계산하면? 만날 때까지 걸리는 시간은 같으므로, L/80 = (240-L)/120 식을 만들 수 있고, 처음 만나는 위치가 L=96m 임을 알 수 있다.

이제, 얼마만의 이동거리(A가 움직이는 거리로 계산하는 것이 나중에 편리)마다 만나는지 계산하기 위해서 400m 트랙을 생각하고 만나면 다시 반대편에서 달려온다고 생각하면 된다.



마찬가지로 마주칠 때까지 움직인 시간이 같으므로,

x/80=(400-x)/120 을 계산하면 160m(A가 움직인 이동 거리)를 계산할 수 있다.

정리하면, A가 96m 이동 후, 160m 이동할 때마다 만나게 됨. 따라서 처음 상황에서 A가 40000m(100바퀴) 이동하면, 96m 이동하고 나서 160m 이동할 때마다 만나는 것이므로, (40000-96)/160 = 249.6 이므로, 총 250번 만나게 된다.

다른 접근 방법...

A가 시계 방향으로 달리고 B가 반시계 방향으로 달리면, B는 서있는 상태에서 A가 더 빨리 달리는 것과 같이 생각할 수 있다.

그렇게 생각하면 A가 시계 방향으로 200m/s로 돌고 B는 정지로 생각할 수 있다. 여기서 A가 80m/s로 100바퀴를 돌았을 때 거리가 40000m 인데. 200m/s로 돌면 80 : 40000=200 : x 로 계산하면 100000m라는 거리가 나온다. 이 의미는 A가 200m/s로 돌았을 때에는 100000m를 도는 것과 같은 시간이라는 의미가 된다.

이렇게 시작하는데. A가 정지되어 있는 B를 만날 때까지 이동한 거리가 240m 이므로 A가 B를 처음 만난 이후 이동해야할 이동거리는 100000-240(m)가 된다. A와 B는 1번 만난 상태로 생각하고, 이제 그 곳에서 시작해서 A가 400m 짜리 트랙을 몇 바퀴를 돌게 되는지를 계산하면 된다. 99760m/400m -> ? 트랙을 돈 횟수.

또 다른 문제 해결 아이디어...

A가 반대로 100번 돌면 마지막에 다시 제자리로 돌아온다. '그럼 도대체 B는 몇 번 A위치를 지나갈까?' 라고 생각하면 B의 속도가 80:120으로 1.5배 빠르므로 B는 150번 지나가게 된다. 그럼 이 둘은 그 동안 몇 번 만났을까?(지나쳤을까?) 250번이 된다.

원형 저수지 위의 트랙을 도는데, 나는 어떤 나무를 10번 봤다. 너는 반대로 돌면서 그 나무 몇 번 봤니? 그 횟수를 알면 서로 마주친 횟수의 결과가 나오는 것인가?

12. 10개의 수들이 20, 40, 70, 10, 80, 30, 100, 50, 60, 90의 순서로 나열되어 있다. 이 수들에 대하여, 임의의 두 수의 교환을 반복 수행하여, 작은 수부터 큰 순서대로 다시 나열하고자 한다. 교환하는 두 수는 인접하지 않아도 된다. 두 수의 교환 횟수의 최솟값은 얼마인가?

① 6 ② 7 ③ 8 ④ 9 ⑤ 10

** 출제의도?

선택 정렬의 이해

** 의미?

먼저, 최솟값의 위치를 찾은 후 자리를 바꾸는 선택정렬 방법에 대한 이해

** 해결예시?

[2 4 7 1 8 3 10 5 6 9]

1 [4 7 2 8 3 10 5 6 9]

1 2 [7 4 8 3 10 5 6 9]

…

1 2 3 4 5 6 7 8 9 10

교환 횟수를 세면 된다.

13. 세 천재 A, B, C가 다음의 사실 (i), (ii), (iii)을 모두 알고 있다.

(i) A, B, C가 가지고 있는 공은 모두 13개이다.
(ii) A, B, C가 가지고 있는 공의 개수는 모두 다르고, 각각 하나 이상의 공을 가지고 있다.
(iii) A는 가장 적은 공을 가지고 있고, C는 가장 많은 공을 가지고 있다.

이때, A, B, C는 자신이 갖고 있는 공의 개수만 확인한 상태에서, 차례로 다음과 같이 말했다.

A: (충분히 생각한 후) B와 C가 각각 몇 개의 공을 가지고 있는지 알 수 없었다.
C: (A의 말을 듣고 충분히 생각한 후) A와 B가 각각 몇 개의 공을 가지고 있는지 알수 없었다.
B: (A와 C의 말을 듣고 충분히 생각한 후) A와 C가 각각 몇 개의 공을 가지고 있는지 알 수 없었다.

A, B, C가 말한 것이 항상 참일 때, B가 가지고 있는 공은 몇 개인가?

① 2　　② 3　　③ 4　　④ 5　　⑤ 알 수 없음

** 출제의도?

관계식 찾기. 논리적 사고. 경우의 수.

** 의미?

주어진 문제/상황에서 관계식을 찾고, 논리적 사고를 이용해 가능한 경우를 선택적으로 제거한다.

** 해결예시?

(i) 에 의하면 A+B+C=13

(ii) 에 의하면 1<=A<B<C<=10(왜냐하면 A, B가 가장 작은 경우 1,2가 되는데 그때 C는 최대 10)

A: 의 말에 의하면 A는 1 또는 2가 되어야 한다. 왜냐하면 3인 경우 B와 C는 4, 6으로 확정되기 때문이다.

C: 의 말에 의하면 C는 7 또는 8이 되어야 한다. 왜냐하면 10인 경우 A,B가 1,2로 확정, 9인 경우 A,B로 합이 4가 되려면 1,3으로 확정된다.

B: 의 말에 의하면 B는 4가 되어야 한다. 왜냐하면 3인 경우 A,C가 2,8로 확정, 5인 경우 1,7로 확정, 6인 경우 불가능하다.

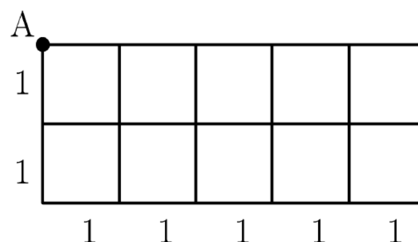
다른 방법: 가능한 수 조합을 먼저 만들어 두고 제거해 가면 된다.

가능 한 경우(찾는 방법? A가 1인 경우, B가 더 크고, C가 더 크면서 13이 되는 조합(경우)만 찾아본다)

1 2 10
1 3 9
1 4 8
1 5 7
2 3 8
2 4 7
2 5 6
3 4 6

예를 들어, A: 의 말에 의하면 3 4 6은 안 된다. 따라서 1또는 2만 가능하다. C: 의 말에 의하면 1 2 10, 1 3 9는 안 된다. 따라서 7또는 8이어야 한다.

14. 다음 도형에서 작은 사각형은 모두 가로길이와 세로길이가 1인 정사각형이다. 이 도형의 점 A에서 출발하여 선분을 따라 움직이면서, 도형의 모든 선분을 지나 A로 다시 돌아오고자 한다. 같은 선분을 두 번 이상 지나도 된다. 지나는 거리의 최솟값은 얼마인가?



① 32 ② 34 ③ 36 ④ 38 ⑤ 40

** 출제의도?

아이디어 확장

** 의미?

오일러 한 붓 그리기와 관련하여 한 붓 그리기가 가능하도록 확장(새로 길 놓기)하는 방법으로 최단 거리 비용 계산하기?

** 해결예시?

(초등 해설 참고) 각 교차점에서의 연결 간선의 개수(그래프에서 정점의 차수)를 적고, 홀수 차수의 정점들을 묶어 모두 짝수로 만드는 가장 적은 연결을 만들고 세면 된다. 어떤 홀수 점들을 어떻게 연결할지가 매우 중요하다. 홀수-홀수 점을 서로 연결하는 것이 기본이지만, 홀-짝-홀 연결에서 좌우를 연결해도 모두 짝수가 될 수 있음을 생각해야 한다.

15. 1부터 30까지의 서로 다른 자연수 30개가 칠판에 쓰여 있다. 이 중에서 임의의 두 수 a, b를 선택한 후 a와 b를 지우고, |a-b|를 칠판에 새로 적는다. 이 과정을 29번 반복하면 칠판에는 숫자 하나만 남게 된다. 1에서 30까지의 수들 중에서

최종적으로 칠판에 남을 수 없는 숫자들을 모두 더하면 얼마인가?

① 0 ② 128 ③ 225 ④ 240 ⑤ 255

** 출제의도?

패턴 분석

** 의미?

단순한 규칙과 상황을 준 후, 데이터가 커졌을 때의 상황에 대한 통찰? 분석?이 필요. 정수론?

** 해결예시?

규칙이나 의미가 잘 떠오르지 않는 경우, 보다 간단한 상황에 대해 생각해 보거나, 가장 작은 상황들을 몇 개 해보면서 규칙을 찾아볼 수 있도록 하는 문제 해결 전략이 필요하다. 가장 작은 경우에 대해서 점차적으로 키워가면서 규칙을 발견할 수 있도록 한다.

1 2 라면? 1만 남는다.

1 2 3 이라면? 0 또는 2만 남길 수 있음. 1은 못 남긴다.

1 2 3 4 라면? 0, 2, 4만 남길 수 있다.

1 2 3 4 5 라면? 1, 3, 5만 남길 수 있다.

1 2 3 4 5 6 라면? 1, 3, 5만 남길 수 있다.

분석해 보면 주어진 집합(1~n)에 들어있는 홀수의 개수가 중요하다는 것을 발견할 수 있다.

왜냐하면 어떻게 빼던지 홀수가 홀 수개가 들어있다면? 마지막에 남는 수는 무조건 홀수가 된다. 반대로 홀수가 짝 수개가 들어있다면 마지막에 남는 수는 무조건 짝 수가 된다.

또한, 홀수로 끝나는 경우(예를 들어 1, 2, 3, 4, 5)에는 마지막 홀수까지 만들 수 있음. 짝수로 끝나는 경우에는 바로 전 홀수까지 만들 수 있다.

그럼 얼마나 남는가? 1~n까지 홀 수의 개수는 올림. ceiling(천장) n/2 즉, $\left\lceil \frac{n}{2} \right\rceil$ 로 표현할 수 있다. 예를 들어 n=5이면 3개, n=6이어도 3개이다.

문제로 돌아가서 n=30이면 홀 수가 15개 이므로 무조건 홀수만 남는다. 따라서, 못 만드는 수는 2, 4, 6, 8, ... , 30 이 된다.

16. 다음 프로그램의 출력 결과는 무엇인가?

```
int a=0;
a=2014/4;
a-=2014/100;
a+=2014/400;
printf("%d\n", a);
```

① 488.395 ② 488.4 ③ 488 ④ 488.39 ⑤ 487

** 출제의도?

C언어 연산자에 대한 이해

** 의미?

나누기 연산자의 특성(정수/정수의 결과는 정수). 복합대입 연산자. 연산 순서. 단순 계산

** 해결예시?

2014를 4로 나눈 몫에 2014를 100으로 나눈 몫을 빼고, 다시 2014을 400으로 나눈 몫을 더한 결과이다.

17. 다음 프로그램의 출력 결과는 무엇인가?

```
int a;
for(a=2014; a>0; a--);
printf("%d\n", a);
```

① -1　② 0　③ 1　④ 2013　⑤ 2014

** 출제의도?
for 반복문에 대한 이해?
** 의미?
for 반복문 실행 후 변수에 저장되어있는 값에 대한 명확한 이해
** 해결예시?
a가 1일 때, a값에 저장된 값을 1 감소시켜서 0로 바뀌고, 다음 조건 검사를 할 때 0>0의 결과가 거짓이므로 a에는 0이 저장되어있는 상태로 for 문이 종료된다.

18. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, a=2, b=5;
for(i=1; i<=100; i++){
  a+=b*i;
}
printf("%d\n", a);
```

① 25250　② 25251　③ 25252　④ 25253　⑤ 25254

** 출제의도?
코드 이해? 패턴 분석? 계산 간략화?
** 의미?
일정 비율로 커지는 수열의 합?
** 해결예시?
처음 초항이 2이고, +5, +10, +15, ... , +500 으로 계산된 결과 출력하기.
2+(5+10+15+ ... + 500) = 2+5*(1+2+3+ ... + 100) = 2+5*5050

19. 다음 프로그램이 5, 4, 3, 2, 1의 순서로 자연수를 출력하기 위해서 빈 칸에 들어갈 코드는 무엇인가?

```
int i=0;
while(i<5){
  printf("%d\n", [      ]);
  i++;
}
```

① i ② i-6 ③ 6-i ④ i-5 ⑤ 5-i

** 출제의도?
패턴 분석. 계산식 만들기
** 의미?
주어진 패턴을 만들 수 있는 계산식 만들기
** 해결예시?
빼는 수를 증가시킨다.

20. 다음 프로그램의 출력은 무엇인가?

```
int i, t=0;
for(i=1; i<=10; i++) t+=i/3;
printf("%d\n", t);
```

① 15 ② 16 ③ 17 ④ 18 ⑤ 19

** 출제의도?
패턴 분석?
** 의미?
반복으로 만들어지는 계산식 패턴을 보고 결과 값 예측하기
** 해결예시?
1/3 + 2/3 + 3/3 + 4/3 + 5/3 + 6/3 + 7/3 + 8/3 + 9/3 + 10/3 의 패턴으로.. 더해지는 값은
0 0 1 1 1 2 2 2 3 3 을 모두 더하면 된다.

21. 다음 프로그램에서 a = a + i; 라는 문장은 몇 번 수행되는가?

```
int i, j, a=0;
for(i=0; i<10; i++){
  for(j=0; j<5; j++) a=a+i;
}
```

① 5 ② 10 ③ 15 ④ 50 ⑤ 150

** 출제의도?
2중 반복의 이해

** 의미?
2중 반복 실행 구조의 실행 횟수에 대한 이해? 계산량 분석?과 연관?
** 해결예시?
바깥 반복문에서 0부터 9까지 10개 선택 된다. 다시 안쪽 반복문에서 각 경우에 대해서 다시 0부터 4까지 5개 선택된다. 가장 안쪽 실행문은 10*5회 실행된다.

22. 다음 중 변수의 이름으로 사용할 수 없는 것은 무엇인가?

① new_day ② new_int ③ summer
④ old-days ⑤ _sum

** 출제의도?
C언어 기초
** 의미?
영문자나 _(언더라인)으로 시작하는 단어들은 변수/함수/배열로 인식된다.
** 해결예시?
- 는 연산자로 사용되기 때문에 변수/함수/배열의 이름에 사용될 수 없다.

23. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, min, a[5]={6,4,2,8,10};
min=a[0];
for(i=1; i<5; i++){
  if(a[i]<min) min=a[i];
}
printf("%d\n", min);
```

① 6 ② 4 ③ 2 ④ 16 ⑤ 30

** 출제의도?
코드 이해. 배열 참조
** 의미?
배열에 저장되어있는 값들 중에서 최솟값 찾기
** 해결예시?
6, 4, 2, 8, 10 중에서 최솟값은?

24. 다음 프로그램의 출력 결과는 무엇인가?

```
int a=20, b=10;
a=a+b;
b=a-b;
a=a-b;
```

```
printf("%d %d\n", a, b);
```

① 10 20　　② 20 20　　③ 10 10

④ 20 10　　⑤ 30 10

** 출제의도?
값 바꾸기의 다른 방법에 대한 이해
** 의미?
일반적으로 임시 변수를 이용해서 두 변수에 저장되어있는 값을 바꾸지만, 임시 변수 없이도 계산으로 두 변수의 값을 바꿀 수 있다는 의미한다.
** 해결예시?
참고 : 계산하는 (순차실행)순서가 매우 중요하다.

25. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, tot=0;
for(i=0; i<5; i++){
 tot=tot+(i-1)*(i+1);
}
printf("%d\n", tot);
```

① 16　　② 25　　③ 26　　④ 28　　⑤ 30

** 출제의도?
코드 이해
** 의미?
???
** 해결예시?

i	0	1	2	3	4	
(i-1)*(i+1)	-1	0	3	8	15	<- 합?

26. 다음 프로그램의 출력 결과는 무엇인가?

```
int a=2, b=3;
while(b>0){
  b--;
  a*=a;
}
printf("%d\n", a);
```

① 256　　② 128　　③ 64　　④ 16　　⑤

** 출제의도?
코드 이해
** 의미?

카운터
** 해결예시?
b 3 2 1 0
4 16 256 X

27. 다음 프로그램의 출력 결과는 무엇인가?

```
int n=130, d=5, count=0;
while(n>0){
  n/=d;
  count+=n;
}
printf("%d\n", count);
```

① 26 ② 31 ③ 32 ④ 161 ⑤ 162

** 출제의도?
코드 이해
** 의미?
130/5... 130/5/5... 130/5/5/5
** 해결예시?
26 + 5 + 1

28. 다음 프로그램의 출력 결과는 무엇인가?

```
void f(int x){
  if(x==0) return;
  f(x/2);
  printf("%d", x%2);
}
int main(){
  f(10);
  printf(" ");
  f(7);
  printf("\n");
  return 0;
}
```

① 100 11 ② 1010 111 ③ 10 7
④ 0101 111 ⑤ 111 111

** 출제의도?

재귀 함수의 이해?
** 의미?
재귀적 방법으로 2진수변환
** 해결예시?
10진수 10은 2진수로 1010, 10진수 7은 2진수로 111 이다.

29. 다음 보기 중에서 가장 큰 값은 무엇인가?

```
int f(int n, int m) {
 if(n==m) return n;
 if(n>m) return f(n-m, m);
 else return f(n, m-n);
}
```

① f(4, 9) ② f(7, 13) ③ f(10, 15)
④ f(12, 6) ⑤ f(12, 8)

** 출제의도?
재귀 함수의 이해?
** 의미?
유클리드 호제법의 재귀 버전. 최대 공약수 찾기
** 해결예시?
(4, 9)의 최대 공약수는 1, (7, 13)의 최대 공약수는 1, (10, 15)의 최대 공약수는 5이다.

30. f(0, 7)의 값은 무엇인가?

```
int c[8]={1,3,7,2,4,1,5,3};
int f(int a, int b){
  int mid;
  if(a==b) return c[a];
  else{
    mid=(a+b)/2;
    return f(a, mid)-f(mid+1, b);
  }
}
```

① -8 ② -5 ③ 0 ④ 5 ⑤ 6

** 출제의도?
재귀 함수의 이해
** 의미?
다중 재귀의 이해

** 해결예시?
두 구역으로 나누어 왼쪽 값에서 오른쪽 값을 빼는 형태. 구역을 계속 나누어간 후 재귀적으로 계산해 올라오면 된다.

31. 다음 보기 중에서 가장 큰 값은 무엇인가?

```
int f(int n){
 if(n==0) return 0;
 return (n%10)+f(n/10)+f(n/100);
}
```

① f(476) ② f(512) ③ f(1091)
④ f(2014) ⑤ f(3021)

** 출제의도?
재귀 함수의 이해?
** 의미?
3자리 10진수에 대해서, 각 자리 수의 합
** 해결예시?
각 자리 숫자의 합을 계산해 보면 된다.

32. 다음 프로그램의 출력 결과는 무엇인가?

```
int n=2014, c=0;
while(n>0){
  c+=!(n&1);
  n=n>>1;
}
printf("%d\n", c);
```

① 0 ② 1 ③ 2 ④ 9 ⑤ 11

** 출제의도?
코드 이해
** 의미?
2진수로 변환시 0의 개수 찾기?
** 해결예시?
10진수 2014를 2진수로 변환하면 11111101110 이다.

33. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[4][4]={{0,3,2,2 },{3,5,8,1},{9,7,4,7},{6,8,1,9}};
```

```
void f(int b, int c){
  int i, k;
  for(i=0; i<4; i++){
    k=a[b][i];
    a[b][i]=a[c][i];
    a[c][i]=k;
  }
}
void g(int b, int c){
  int i, k;
  for(i=0; i<4; i++){
    k=a[i][b];
    a[i][b]=a[i][c];
    a[i][c]=k;
  }
}
int main(){
  int i;
  f(0, 2);
  g(1, 2);
  f(3, 1);
  for(i=0; i<4; i++) printf("%d ", a[i][i]);
  printf("\n");
  return 0;
}
```

① 0 5 4 9　② 2 7 5 4　③ 4 9 5 0
④ 9 1 3 1　⑤ 9 8 8 2

** 출제의도?
함수의 이해
** 의미?
2차원 배열의 줄끼리 바꾸기, 열끼리 바꾸기
** 해결예시?
f(b, c) : b줄, c줄 서로 바꾸기
g(b, c) : b열, c열 서로 바꾸기

34. 다음 프로그램의 출력 결과는 무엇인가?

```
int a, i;
a=2014;
```

```
for(i=0; i<4; i++) a|=(a>>(1<<i));
a++;
printf("%d\n", a);
```

① 2014 ② 2023 ③ 2047 ④ 2048 ⑤ 2049

** 출제의도?

코드 이해, 비트단위 시프트 연산자의 이해, 비트단위 논리 연산자의 이해, 연산자 우선순위 이해, 2진 표현의 이해.

** 의미?

2진 표현에 대한 비트단위 연산자의 계산.

** 해결예시?

이 문제는 특별한 방법이나 아이디어는 없다. 실행되는 순서대로 비트 연산을 해보는 과정에서 답을 찾을 수 있다.

비트 연산을 따라가기 위해 2014를 2진수로 변환하면

2014 : 00000000 00000000 00000111 11011110

>>1 : 00000000 00000000 00000011 11101111

위 두 값을 비트 or(|) 연산하면

00000000 00000000 00000111 11111111

>>2 : 00000000 00000000 00000001 11111111

위 두 값을 비트 or(|) 연산하면

00000000 00000000 00000111 11111111

>>4 : 00000000 00000000 00000000 01111111

위 두 값을 비트 or(|) 연산하면

00000000 00000000 00000111 11111111

...

값이 변하지 않음....

여기에 마지막에 1을 더하면

00000000 00000000 00001000 00000000

인데, 이 값을 10진수로 변환한다.

35. f(4, 0)의 결과값은 무엇인가?

```
int f(int a, int b){
  if(a==0 && b==0) return 1;
  if(b<-a || b>a) return 0;
  return f(a-1, b)+f(a-1, b-1)+f(a-1, b+1);
}
```

① 0 ② 1 ③ 7 ④ 16 ⑤ 19

** 출제의도?

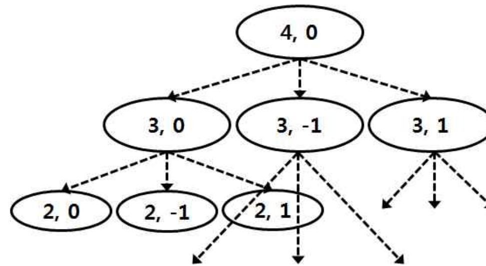
재귀함수의 이해

** 의미?

재귀함수의 실행 과정을 동적테이블을 채우는 방법으로 변환시켜 단순화시키고 계산량을 줄일 수 있다.

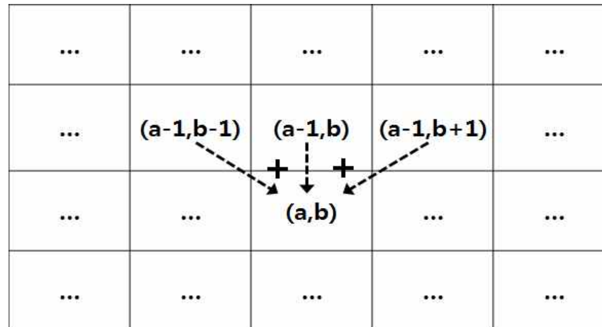
** 해결예시?

처음 호출되는 함수 f(4, 0)을 초기 상태라고 생각하고, 재귀함수에 의해 분기되는 상황을 트리 형태로 그려볼 수 있다.



하지만, 그렇게 하면 경우의 수가 너무 많아지기 때문에 깊이를 더 내려가면 매우 많은 상태가 만들어지기 때문에 계산과 예측이 매우 어렵게 된다. 현재 주어진 재귀함수의 호출은 하나의 상태가 다시 3개의 상태로 분기되기 때문에 3진 트리 그리기와 같은 정도의 상태들이 생기게 되는데, 그렇게 살펴보면 같은 상태가 중복되는 경우들이 많아지면서 계산량이 매우 크게 증가하게 된다. 대략적 계산량 O(n3)이다.

상태가 분기되는 상황을 거꾸로, 이전 상태의 값을 참조해서 다른 상태값을 만들어가는 방법으로 생각을 변환하고 그러한 상태를 표 채우기(동적 프로그래밍) 형태로 바꾸어 생각할 수 있다. 이렇게 형태를 바꾸어 생각하면 중복되는 상황을 고려하지 않고, 어떤 상태에 대해서 1번만 계산하고 생각할 수 있는 형태로 계산량이 감소하게 된다.



주의 사항!! : 마이너스 상태에 대한 직관과 처리가 반드시 필요하다.

가장 윗줄의 왼쪽 셀부터 채워나가는데, 논리식을 보고 순서대로 아래로 채워 나가면서 (4, 0)의 값이 어떻게 만들어지는지 확인한다.

a \ b	-4	-3	-2	-1	0	1	2	3	4	5	6
0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	1	1	1	0	0	0	0	0
2	0	0	1	2	3	2	1	0	0	0	0
3	0	1	3	6	7	6	3	1	0	0	0
4	1	4	10	16	**19**	?	?	?	?	?	?

표를 어떻게 채워나가는지 이해하는 것이 가장 중요하다. (0, 0)의 값이 1인 것으로 시작하면서 이해해 보자.

36. 다음과 같은 함수 f가 있을 때, (a)와 (b)에 들어가는 값 중 다른 결과를 출력하는 문자열 쌍은 무엇인가?

```
int f(char* a, char* b){
  if((*a)=='\0'|| (*b)=='\0' || (*a)!=(*b)){
    int res=(*a)-(*b);
    if(res<0) return -1;
    if(res>0) return 1;
    return 0;
  }
  return f(a+1, b+1);
}

int main(){
  char A[]="(a)";
  char B[]="(b)";
  printf("%d\n", f(A, B));
  return 0;
}
```

① (a) abcd (b) tuvwx
② (a) x (b) yz
③ (a) ABCD (b) AUYZ
④ (a) UIO (b) UFO
⑤ (a) QWERTY (b) YTREWQ

** 출제의도?
함수에 배열 전달?, 재귀 함수의 이해?

** 의미?
재귀적 문제 해결 방법 이해!
주어진 코드는 재귀 함수에 두 배열의 위치를 전달하고, 재귀적으로 한 문자씩 판단해 나아가는 재귀적 코드이다. 코드의 내용은 결과적으로 전달된 두 문장(문자열)의 사전 순 배열을 위한 비교 결과를 돌려주는 함수를 재귀적으로 구현한 것이다.

** 해결예시?
두 문장의 사전 순(영어 사전에 나열되는 순서) 결과가 다른 것은?

37. 다음 보기 중 결과값이 1인 것은 무엇인가?

```
int f(int a){
  int i, r;
  r=0;
  for(i=1; i<=a; i++) if(a%i==0) r=1-r;
  return r;
}
```

① f(121)　② f(123)　③ f(127)
④ f(128)　⑤ f(255)

** 출제의도?
코드 이해, 함수의 이해
** 의미?
완전제곱수 찾기?
** 해결예시?
어떤 수의 약수의 개수가 홀수 개이어야만 r이 1이 되므로, 완전 제곱수 이어야만 한다.

38. 다음 프로그램의 출력 결과는 무엇인가?

```
char enc[12] = "mississippi";
int i, j;
for(i=0; i<3; i++){
  for(j=i; j<11; j+=3) printf("%c", enc[j]);
}
printf("\n");
```

① iiiissssppm　② ippississim　③ mississippi
④ msspisspiii　⑤ msspiiiissp

** 출제의도?
코드 이해
** 의미?
3개의 부분으로 쪼갠 후, 각 부분별로 하나씩 출력한다.
** 해결예시?
mis sis sip pi
맨 앞에 문자 하나씩 출력하면 mssp
두 번째 문자 하나씩 출력하면 iiii
세 번째 문자 하나씩 출력하면 ssp

39. f(5)의 결과값은 무엇인가?

```
int f(int n){
  int sum, i;
  if(n==0) return 1;
  sum=0;
  for(i=0; i<n; i++) sum+=f(i)*f(n-i-1);
  return sum;
}
```

① 40　② 41　③ 42　④ 43　⑤ 44

** 출제의도?
함수의 이해, 재귀 함수의 이해
** 의미?
재귀 함수 안에 여러 개의 재귀가 호출되는 다중 재귀.... 조합? (n-1)개 만들기?
** 해결예시?
일일이 따라갈 수 있을까? 가능은 하겠지만 f(5)가 아니라 f(9)만 되어도 완전 불가능하다.
재귀 함수의 호출 형태를 표 채워나가기 형태로 바꿔 원하는 값f(5)가 나올 때까지 쌓아올려가 보자.

n	0	1	2	3	4	5	
f(n)	1	1	2	5	14	...	

각 f(n)의 값들의 의미를 순서대로 풀어써보면 아래와 같다는 것을 이해하고 위와 같은 표를 만들 수 있다.
f(0) = 1
f(1) = f(0)*f(0) = 1
f2) = f(0)*f(1)+f(1)*f(0) = 2
f(3) = f(0)*f(2)+f(1)*f(1)+f(2)*f(0) = 5
f(4) = ...
f(5) = ...

재귀함수가 실행되면서 호출되는 형태를 이해하기 위해서 가장 작은 것부터 한 단계씩 형태를 풀어보면 어떤 패턴을 찾아낼 수 있는데, 처음에는 이러한 형태에 대한 이해가 쉽지는 않다. 하지만, 곰곰이 생각해보면서 이해할 수 있도록 해야 한다.
특히 이런 형태의 하향식 재귀는 매우 간단한 상향식 재귀 형태로 변활 될 가능성이 충분히 있다.
** 하향식 재귀 : f(n) 이 그 보다 작은 f(n-...) 들을 호출하는 형태. 출력할 값은 보통 f(n) 또는 f(n-1)을 출력한다. 상향식 재귀 f(n) 이 f(n+1)... 등을 호출하는 형태. 출력할 값은 보통 f(0) 또는 f(1)을 출력한다.

40. 다음 프로그램의 출력 결과는 무엇인가?

```
char str[10]="awzydebcx";
char tmp;
int i;
for(i=0; i<9; ++i){
  if(str[i]<str[8-i]){
    tmp=str[i];
    str[i]=str[8-i];
    str[8-i]=tmp;
  }
}
printf("%s \n", str);
```

① acbedxwzy ② awzydebcx ③ abcxdewzy
④ acbedyzwx ⑤ xcbedyzwa

** 출제의도?
코드 이해
** 의미?
배열에 저장되어있는 8글자 문자열의 가운데를 중심으로 좌우를 쌍으로 비교하면서 자리 교환.
** 해결예시?

n	0	1	2	3	4	5	6	7	8	9
str(n)	a	w	z	y	d	e	b	c	x	

(0, 8)을 비교하고, (1, 7)을 비교하고. 계속 비교하면서 자리를 바꾸면 된다.

41. 다음 프로그램의 출력 결과는 무엇인가?

```
int n, i, b;
char str[10];
char hex[]="0123456789ABCDEF";
n=2014, i=0;
for(b=1; b<=n; b*=16) str[i++]=hex[(n/b)%16];
str[i]='\0';
printf("%s\n", str);
```

① 7DE ② ED7 ③ 2014 ④ ED70 ⑤ 07DE

** 출제의도?
코드 이해
** 의미?
배열+반복 실행 구조를 이용한 진법 변환 코드. 10진수 2014를 16진수로 변환하여 출력한다. 단, 진법 변환된 결과가 역(반대)로 출력된다.
** 해결예시?
10진수 2014를 바로 16진수로 계산할 수도 있지만... 2진수로 먼저 변환하고 난 후 4개씩 묶어 16진수로 변환하는 것이 편리하다.
10진수 2014는 2진수로 11111011110 이다. 소수점을 중심으로 4개씩 묶으면 111 1101 1110 이고, 이를 16진수의 문자(배열에서 설정되어있음)로 변환하면 16진수로 7DE 가 된다. 거꾸로 출력되어야 한다.

42. 아래 프로그램에서 함수 f는 a<=x<=b 이고, c<=x<=d인 정수 x의 개수를 반환한다. (단, -1000 <= a, b, c, d <= 1000 는 항상 만족한다.

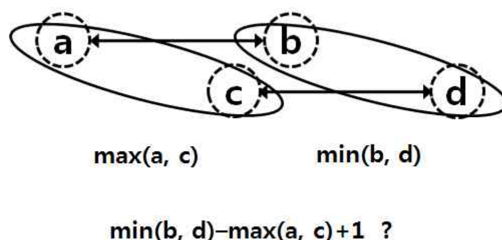
```
int min(int a, int b){
  if(a<b) return a;
  return b;
}
int max(int a, int b){
  if(a>b) return a;
  return b;
}
int f(int a, int b, int c, int d){
  return [(a)](0, [(b)](b, d)-[(c)](a, c)+1);
}
```

① (a) max (b) min (c) max　　② (a) min (b) max (c) min
③ (a) min (b) min (c) max　　④ (a) max (b) max (c) max
⑤ (a) max (b) min (c) min

** 출제의도?
코드 이해. 함수의 이해

** 의미?
문제에서의 조건에서 어떤 두 범위에 들어가는 공통 영역을 찾아내는 것과 유사한 의미. 두 부등식을 만족하는 정수값의 개수를 찾아내는 f( ) 함수를 설계해야 한다. 사용할 수 있는 것은 작은 값을 출력하는 min( )과 큰 값을 출력하는 max( ) 뿐이다.
결국 min( ), max( ) 함수만을 이용해서 주어진 두 범위의 교집합에 속하는 정수 값의 개수를 찾아내는 방법에 대한 이야기이다.

** 해결예시?
상황이 잘 이해가 되지 않을 수 있으므로, 간단한 그림을 그린 후 f( ) 함수의 호출과 비교해 보는 방법으로 찾아낼 수 있다. 이런 식을 창의적으로 만들어내는 것도 의미가 있지만, 주어진 틀을 보고 거꾸로 생각해 보면서 그렇게 만들어낼 수 있는 경험과 능력을 키울 수 있다.
문제에서 a<= x <= b 이고, c <= x <= d 인데, -1000 <= a, b, c, d <= 1000 이므로, a<b, c<d 인 관계를 찾아낼 수 있다. 그리고 아래와 같은 그림을 통해 필요한 식을 생각해내면 된다.



43. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[10]={3,0,1,2,3,2,4,5,9,8};
int result, i;
result=0;
for(i=0; i<10; i++){
  int p, q, len;
  p=i, q=a[i], len=1;
  while(p!=q){
    p=a[p];
    q=a[a[q]];
  }
  q=a[q];
  while(p!=q){
    p=a[p];
    q=a[a[q]];
    len++;
  }
  if(result<len) result=len;
}
printf("%d\n", result);
```

① 1 ② 2 ③ 3 ④ 4 ⑤ 5

** 출제의도?

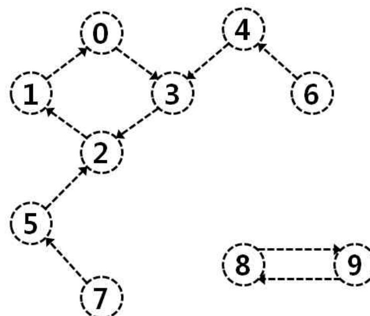
코드 이해. 배열의 참조(인덱싱)/2중 참조(더블 인덱싱) 이해.

** 의미?

참조와 2중 참조로, 계속해서 값이 변해가는 형태이다. 배열의 참조를 그래프로 변환해 이해?

** 해결예시?

1차원 배열의 계속적인 참조를 방향 그래프 형태로 표현한 후 그 과정을 살펴보면 매우 간단하게 이해할 수 있게 된다. 참조 번호와 그에 따른 값을 서로 연결하는 방향 그래프를 생각하고 그림을 그려보면 아래와 같이 그릴 수 있다.



p=a[p] 는 한 단계 이동, q=q[q[a]] 는 두 단계 이동이라고 생각하면 주어진 코드는 어떤 점에서 시작해서, 한 턴에 p는 한 단계, q는 두 단계 이동을 하면서 한 번 만난 후 다시 만날 때까지의 최대 턴 수를 구하는 의미가 된다. 그래프를 따라가면서 시작 점, 만나는 점, 다시 만날 때까지 걸리는 턴을 정리하면

시작점, 만나는 점, 다시 만나는 점, 걸리는 턴...

0	1	2	3
1	2	3	3
2	3	0	3
3	0	1	3
4	1	2	3
5	0	1	3
6	2	3	3
7	1	2	3
8	8	8	2
9	9	9	2

처음에 1턴을 먼저 잡고 시작하니 최대 4턴!

44. 다음과 같은 모양을 출력하려고 한다.

```
 1  2  3  4  5
16 17 18 19  6
15 24 25 20  7
14 23 22 21  8
13 12 11 10  9
```

다음은 위와 같은 모양을 출력하기 위한 프로그램이다.

```
int dir[4][2]={ (a) };
int a[7][7]={
  {-1,-1,-1,-1,-1,-1,-1},
  {-1, 0, 0, 0, 0, 0,-1},
  {-1, 0, 0, 0, 0, 0,-1},
  {-1, 0, 0, 0, 0, 0,-1},
  {-1, 0, 0, 0, 0, 0,-1},
  {-1, 0, 0, 0, 0, 0,-1},
  {-1,-1,-1,-1,-1,-1,-1}};
int main(){
  int i, j, r, c, d, nr, nc;
  r=c=1, d=0;
  for(i=1; i<=25; i++){
    a[r][c]=i;
    nr=r+dir[d][0];
    nc=c+dir[d][1];
    if(a[nr][nc]!=0) d=(d+1)%4;
```

```
    r+=dir[d][0];
    c+=dir[d][1];
  }
  for(i=1; i<=5; i++){
    for(j=1; j<=5; j++) printf("%3d", a[i][j]);
    printf("\n");
  }
  return 0;
}
```

다음 중 (a) 에 들어갈 내용으로 알맞은 것은?

① {0,1 }, {1,0 }, {0,-1 }, {-1,0 }   ② {0,0 }, {0,0 }, {0,0 }, {0,0 }
③ {0,1 }, {0,-1 }, {1,0 }, {-1,0 }   ④ {1,1 }, {-1,-1 }, {1,-1 }, {-1,1 }
⑤ {1,1 }, {1,1 }, {1,1 }, {-1,1 }

** 출제의도?
코드 이해
** 의미?
배열 채우기, 달팽이 배열 만들기. 이해
** 해결예시?
배열을 달팽이 모양으로 채워간다고 할 때, 다음 줄(nr)이 다음에 어떻게 변해야하는지, 다음 열(nc)이 다음에 어떻에 변해야하는지를 방향 배열 dir[ ][ ]을 만들어 계산하고 있다. 벽이나 0이 아닌 수에 닿으면 다음 방향으로 방향을 바꾸는 형태의 코드이다.
어떤 점에서 다음 방향으로의 변화가 어떤 순서로 되어야 할까? 오른쪽, 아래쪽, 왼쪽, 위쪽이다. 이렇게 변하는 변위(방향)을 dir[ ][ ] 배열에 순서대로 기록해 두고 참조하면 된다.

45. 다음 프로그램을 수행하였을 때, 5번째로 출력되는 숫자는 얼마인가?

```
int a[10]={1,8,6,1,4,5,3,7,9,0};
int i, j, tmp;
for(i=0; i<9; i++){
  for(j=0; j<9; j++){
    if(a[j]>a[j+1]){
      tmp=a[j];
      a[j]=a[j+1];
      a[j+1]=tmp;
    }
  }
  printf("%d\n", a[8]);
}
```

① 0　　② 1　　③ 2　　④ 8　　⑤ 9

** 출제의도?
코드 이해
** 의미?
버블 정렬
** 해결예시?
배열에 저장되어있는 값을 정렬하면 0 1 1 3 4 5 6 7 8 9 이다. a[8]의 값은?

46. 다음 프로그램의 출력 결과는 무엇인가?

```
int count[101];
int i, j;
for(i=1; i<=100; i++) count[i]=0;
  for(i=1; i<=100; i++){
    for(j=i+i; j<=100; j=j+i) count[j]++;
}
printf("%d\n", count[100]);
```

① 4　　② 6　　③ 8　　④ 10　　⑤ 100

** 출제의도?
코드 이해
** 의미?
에라토스테네스의 체와 비슷한 방법을 이용한 약수의 개수 찾기
** 해결예시?
배열의 어떤 위치의 값을 +1씩 증가시키는데, 형태를 살펴보면 어떤 배수가 될 때 증가시키고 있다. 따라서 각 배열에 저장되어있는 값은 결국 그 값에 대한 약수의 개수와 관련이 된다.
count[100] 의 값은 1, 2, 4, 5, 10, 20, 25, 50 일 때 증가된다. 100일 때에는 증가되지 않고 종료한다. j=i+i.

47. 정수형 배열 arr에서 최솟값을 찾는 코드를 다음과 같이 구현하였다. 빈 칸 (a)와 (b)에 알맞은 구문은?

```
int min_value(int* arr, int idx) {
  int p=0;
  if(idx==0) return  (a) ;
  p=min_value(arr, idx-1);
  if (  (b)  ) return p;
  return arr[idx];
}
```

```
int main() {
  int arr[8]={5,3,9,4,2,5,7,6};
  printf("%d\n", min_value(arr, 7));
  return 0;
}
```

출력 결과
2

① (a) arr[idx] (b) arr[idx] == p
② (a) p (b) p
③ (a) arr[idx] (b) arr[idx] < p
④ (a) p (b) p < arr[idx]
⑤ (a) arr[idx] (b) p < arr[idx]

** 출제의도?
코드 이해, 함수 이해, 배열 전달, 재귀함수의 의해, 재귀적 문제해결
** 의미?
배열에 저장되어있는 데이터 집합에서 최소값을 찾기 위한 재귀적 방법 구현.
** 해결예시?
쉽게 풀어쓰면 현재 보다 하나 작은 집합(0번~(n-1)번)에서 구한 가장 작은 값과 n번 데이터 중에서 작은 값을 리턴하는 재귀적 문제해결 방식의 최소값 찾기이다. 이해하고 빈칸 채워 넣기 하면 된다.

48. 다음 프로그램을 실행할 때, 6번째로 출력되는 값은 무엇인가?

```
int Adj[8][8]={
  {0,0,0,0,0,0,1,0},
  {0,0,0,1,0,0,1,1},
  {0,0,0,0,0,1,0,0},
  {0,0,0,0,1,1,0,0},
  {0,0,1,0,0,0,0,0},
  {0,0,0,0,0,0,0,0},
  {0,0,0,0,1,0,0,0},
  {0,1,0,0,0,0,0,0}};
int visited[8]={0,0,0,0,0,0,0,0};
void f(int v, int n){
  int w;
  visited[v]=1;
  for(w=0; w<n; w++){
    if(Adj[v][w]==1 && !visited[w]) f(w, n);
  }
```

```
  printf("%d\n", v);
}
int main(){
  int u;
  for(u=0; u<8; u++) if(!visited[u]) f(u, 8);
  return 0;
}
```

① 1 ② 2 ③ 3 ④ 4 ⑤ 6

** 출제의도?

코드 이해, 재귀 함수의 이해, 체크 배열의 이해

** 의미?

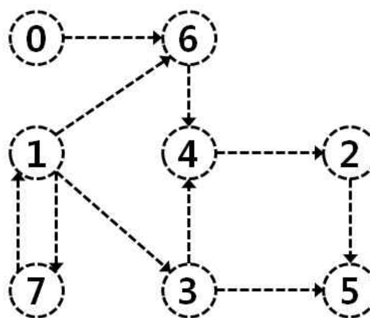
2차 배열로 표현된 그래프의 인접 행렬. 그래프의 정점을 방문(체크)하며 탐색하는 그래프 탐색 관련의 이해. 주어진 초기 배열을 그래프에서 정점과 각 정점 사이의 연결 관계를 표현하는 인접행렬로 인식하고, 재귀 함수에 의해 진행되는 의미를 파악함. 정점을 방문했을 때 체크 배열(visited[ ])에 체크함으로서 한 번 방문한 정점은 다시 방문하지 않도록 처리하고 있다.

재귀 함수의 진행은 현재 정점에서 방문하지 않았던 연결된 정점들을 정점 번호에 따라 순서대로 방문하는데, 더 이상 방문할 수 없을 경우, 즉 막다른 정점이면 그 정점의 번호를 출력한다.

한편, 값(정점 번호)을 출력하는 부분은 어떤? 탐색의 형태로 코딩이 되었는데, 예를 들어 정점의 방문 순서가 0 6 4 2 5 이었고 5번 정점에서 더 이상 갈 수 없는 막다른 정점이라면 출력은 5 2 4 6 0 으로 방문한 정점의 번호가 거꾸로 출력되는 형태이다.

그래프의 깊이 우선 탐색의 결과를 출력하는 것과 같다.

** 해결예시?



0번 정점에서 탐색/방문을 시작하면 5번 정점에서 끝남(막다른 정점)

탐색 경로는 0 6 4 2 5 가 되고 출력되는 값은 5 2 4 6 0이 된다.

1번 정점에서 탐색/방문을 시작하면, 탐색 우선순위!!!(주의)는 1 다음에 3다음에 7이 되는데, 3이 막다른 정점이니 출력, 그 다음 7이 막다른 정점이니 출력, 거꾸로 올라가서 1출력을 하기 때문에 3 7 1이 출력된다.

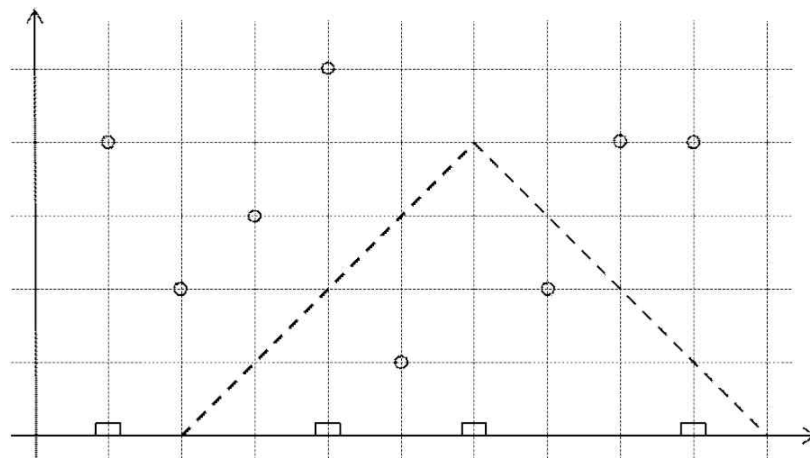
전체적으로 출력된 값을 보면 5 2 4 6 0 3 7 1 이 된다.

[49-50] 다음과 같은 문제를 해결하기 위해 프로그램을 작성하였다. 물음에 답하시오.

KOI 사냥터에는 N마리의 동물들이 각각 특정한 위치에 살고 있다. 사냥터에 온 사냥꾼은 일직선 상에 위치한 M 개의 사대(총을 쏘는 장소)에서만 사격이 가능하다. 편의상, 일직선을 x-축이라 가정하고, 사대의 위치를 x1, x2, ... , xM은 x-좌표 값이라고 하자. 각 동물이 사는 위치는 (a1, b1), (a2, b2), ... , (aN, bN)과 같이 x, y-좌표 값으로 표시하자. 동물의 위치를 나타내는 모든 좌표 값은 양의 정수이다.

사냥꾼이 가지고 있는 총의 사정거리가 L이라고 하면, 사냥꾼은 한 사대에서 거리가 L보다 작거나 같은 위치의 동물들을 잡을 수 있다고 한다. 단, 사대의 위치 xi와 동물의 위치 (ai, bj)간의 거리는 |xi – ai|+bj로 계산한다.

예를 들어, 아래의 그림과 같은 사냥터를 생각해 보자. (사대는 작은 사각형으로, 동물의 위치는 작은 원으로 표시되어 있다.) 사정거리 L이 4라고 하면, 점선으로 표시된 영역은 왼쪽에서 세 번째 사대에서 사냥이 가능한 영역이다.



사대의 위치와 동물들의 위치가 주어졌을 때, 잡을 수 있는 동물의 수를 출력하는 프로그램을 작성하시오.

입력 형식

표준입력의 첫 줄에는 사대의 수 M(1 <= M <= 100,000), 동물의 수 N(1 <= N <= 100,000), 사정거리 L(1 <= L <= 1,000,000,000)이 빈칸을 사이에 두고 주어진다. 두 번째 줄에는 사대의 위치를 나타내는 M개의 x-좌표 값이 빈칸을 사이에 두고 양의 정수로 주어진다. 이후 N개의 각 줄에는 각 동물의 사는 위치를 나타내는 좌표 값이 x-좌표 값, y-좌표 값의 순서로 빈칸을 사이에 두고 양의 정수로 주어진다. 사대의 위치가 겹치는 경우는 없으며, 동물들의 위치가 겹치는 경우도 없다. 모든 좌표 값은 1,000,000,000보다 작거나 같은 양의 정수 이다.

출력 형식

출력은 단 한 줄이며, 잡을 수 있는 동물의 수를 음수가 아닌 정수로 출력한다.

입력과 출력의 예1

입력

```
4 8 4
6 1 4 9
7 2
3 3
4 5
5 1
2 2
1 4
8 4
9 4
```

출력

```
6
```

입력과 출력의 예2

입력

```
1 5 3
3
2 2
1 1
5 1
4 2
3 3
```

출력

```
5
```

프로그램

```
#include <stdio.h>
#define NMAX 100000
#define MMAX 100000
int M, N, L;
int Shoot[MMAX], AnimalX[NMAX], AnimalY[NMAX];
int myAbs(int d){
  if(d<0) return -d;
  return d;
```

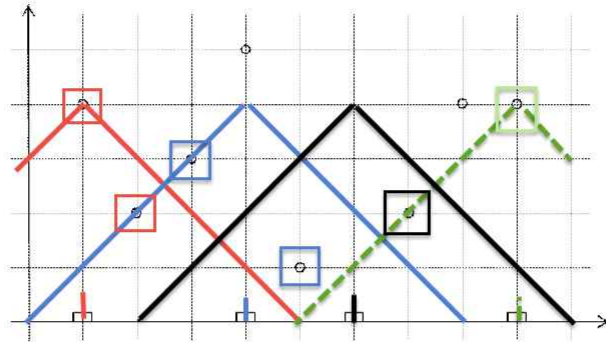
```
}
int main() {
  int i, j, result = 0;
  scanf("%d %d %d", &M, &N, &L);
  for(i=0; i<M; i++) scanf("%d", &Shoot[i]);
  for(i=0; i<N; i++) scanf("%d %d", &AnimalX[i], &AnimalY[i]);

  for(i=0; i<N; i++){
    for(j=0; j<M; j++) {
      if ( [   (a)   ] <=L) {
        result++;
        [   (b)   ] ;
      }
    }
  }
  printf("%d \n", result);
  return 0;
}
```

49. 다음 중 (a)와 (b)에 들어갈 내용으로 알맞은 것은?

① (a) myAbs(Shoot[j] - AnimalX[i]) + AnimalY[i]　(b) break
② (a) myAbs(Shoot[j] - AnimalX[i] + AnimalY[i])　(b) break
③ (a) myAbs(Shoot[j] - AnimalX[i] + AnimalY[i])　(b) continue
④ (a) Shoot[j] - AnimalX[i] + AnimalY[i]　(b) break
⑤ (a) Shoot[j] - AnimalX[i] - AnimalY[i]　(b) continue

** 출제의도?
실제 정보올림피아드 경시 실기 대회 문제 풀이!
** 의미?
출제 되었지만, 많이 풀지 못했던 문제 등. 실제 제시된 문제에 대한 예시 해법을 제시하고 이해할 수 있는 지 평가하는 의미를 가진다.
** 해결예시?
첫 번째 예시 입출력 데이터는 그림의 상태를 의미하는데.. (8, 4)의 동물은 4개의 사대에서 잡을 수 없음. 잡을 수 있는 동물은 6마리이다. 코드 형태를 보니 각 사대에서 동물을 잡을 수 있으면 잡고 카운팅을 올리고, 어떤 처리를 하고 있다.
예시 그림을 통해 분석해 보면, 어떤 사대에서 잡을 수 있는 동물은 사대를 기준으로 가로로 이동해야하는 격자의 수+세로로 이동해야하는 격자의 수의 합과 같거나 작으면 잡을 수 있다.



각 사대에서 어떤 동물까지의 이동거리(격자 거리의 합)는 그 사대의 x좌표에서 동물의 x좌표를 뺀 값의 절대값에 동물의 y좌표의 합으로 계산할 수 있다.

그리고 어떤 동물이 어떤 사대에서 잡힐 수 있는 지만 판단하면, 다른 사대에서 잡을 수 있는지 다시 판단해 볼 필요가 없이 다음 동물에 대해서 순서대로 확인하면 된다.

계산량은 많지만, 각 동물의 관점에서 가능한 모든 경우를 찾아 카운팅 하는 방법이다.

50. 입력이 다음과 같이 주어졌을 때, 위 프로그램의 출력은 무엇인가?

```
6 4 5
1 9 10 6 8 11
2 3
6 5
3 3
8 6
```

① 1 ② 2 ③ 3 ④ 4 ⑤ 5

** 출제의도?

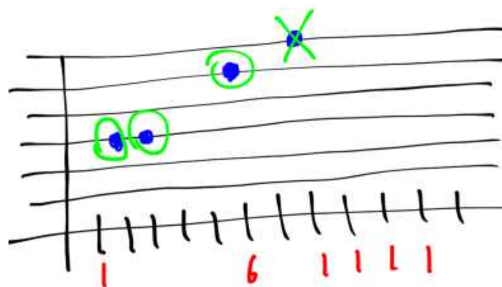
데이터를 의미로 이해하기

** 의미?

문제 상황을 표현하는 데이터를 보고, 거꾸로 문제 상황으로 환원해서 이해할 수 있는지 평가?

** 해결예시?

데이터를 보고 직접 그린 후, 그림을 통해 답을 찾아내면 된다



2015 시도예선 초등부 문제

1. 3원짜리 동전 5개와 7원짜리 동전 5개가 있다. 다음 중에서 이 동전들을 사용하여 지불할 수 없는 금액은?

① 16원 ② 17원 ③ 18원 ④ 19원 ⑤ 20원

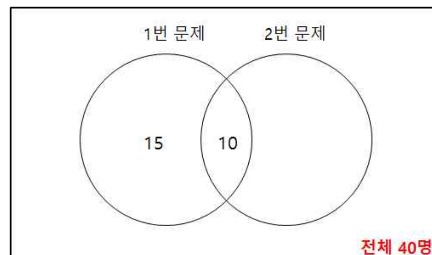
** 출제의도?
가능한 경우의 수 찾기
** 의미?
가능한 경우의 수를 효과적으로 찾아내기
** 해결예시?
보기가 20이하이고 동전이 5개 이므로 20이하 3의 배수(0~15)와 7의 배수(0~14)의 조합을 활용해 지불할 수 있는 금액을 찾은 후 지불할 수 없는 금액을 찾는다.

3의 배수 / 7의 배수	0	3	6	9	12	15
0	0	3	6	9	12	15
7	7	10	13	16	19	X
14	14	17	20	X	X	X

2. 40명의 학생들에게 두 문제를 내어 풀게 했더니 1번 문제를 푼 학생은 정확히 25명이었으며 두 문제를 모두 푼 학생은 정확히 10명이었다. 2번 문제를 푼 학생의 수의 최솟값과 최댓값은 각각 얼마인가?

① 0, 10 ② 5, 25 ③ 10, 25 ④ 20, 25 ⑤ 25, 40

** 출제의도?
집합
** 의미?
주어진 조건으로 가능한 최대, 최소 찾기
** 해결예시?
전체 학생이 40명이고, 1번 문제를 푼 25명 중 2번 문제를 함께 푼 학생이 10명이므로 2번 문제를 푼 학생의 최솟값은 10명, 최댓값은 25명이 된다.



3. 빨간 공 a개와 파란 공 b개가 들어있는 자루가 있다. 이 자루에서 빨간 공 5개를 빼내었더니, 자루에 들어 있는 빨간 공의 개수가 정확히 자루에 들어 있는 전체 공의 개수의 $\frac{3}{5}$이 되었다. a와 b는 모두 1 이상이라고 할 때에, 가능한 a의 최솟값은?

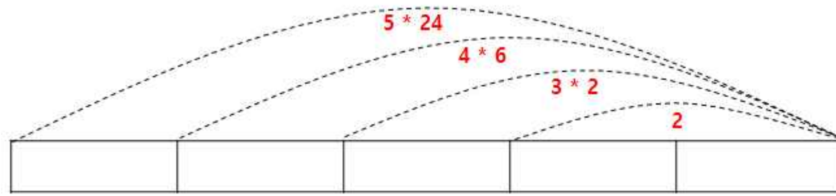
① 5 ② 8 ③ 11 ④ 14 ⑤ 20

** 출제의도?
계산식 만들기
** 의미?
문제에서 주어진 조건들을 수식화 하고, 최솟값 찾아내기
** 해결예시?
1) 해법 1(수식 이용)
수식을 세워서 계산을 하는데, 전체 공의 개수는 $a+b$이므로, $\frac{a-5}{(a-5)+b}=\frac{3}{5}$의 식을 세워서 전개한다. $2a=3b+10$의 식으로 정리되는데, a가 자연수가 되기 위에서는 b가 2이면 가능하므로, a는 8이어야 한다.
2) 해법 2(비율 이용)
- 빨간공 5개를 뺀 후 빨간공의 비율이 3/5, 파란공의 비율이 2/5이다.
- 그럼 앞서 뺀 5개를 빨간공의 비율에 더하면 5/5 + 3/5이므로, 비율이 8/5가 된다.
- 따라서 빨간공은 8/5, 파란공은 2/5의 비율을 차지한다.
- 빨간공과 파란공의 비율을 이용해 최소의 개수를 구하려면 각 비율에 5를 곱하면 되므로 빨간공의 최솟값을 구할 수 있다.

4. 1부터 5까지 숫자를 단 한 번씩만 써서 만든 다섯 자리 수들을 가장 작은 수부터 커지는 순서로 나열했을 때, 50번째에 오는 수는?

① 31245 ② 31254 ③ 32145 ④ 32154 ⑤ 32415

** 출제의도?
순열만들기
** 의미?
순열로 만들어지는 가능한 경우들 정렬하기
** 해결예시?
2자리 일 때 경우는 2개, 3자리 일 때 경우는 6개(3자리 2)
4자리 일 때 경우는 24개(4자리 6), 5자리 일 때 경우는 120개(5자리 24) 이다.
-> 맨 앞자리가 1인 경우 24개, 2인 경우 24개이므로
50번째 오는 수는 맨 앞자리가 3인 수 중 두 번째로 큰 수인 31254 이다.



5. 1부터 999까지의 자연수 중에서 101 또는 522와 같이 같은 숫자를 2개 이상 포함하는 자연수는 모두 몇 개인가?

① 259　　② 260　　③ 261　　④ 262　　⑤ 263

** 출제의도?
나누어 생각하기?
** 의미?
구간을 나누어 체계적으로 카운트하기
** 해결예시?
11* -> 10가지
1*1 -> 9가지(111이 위 경우와 중복되므로 제외)
*11 -> 9가지(111이 위 경우와 중복되므로 제외)
=> (10+9+9)*9 = 252
*00 -> 9가지
=> 252 + 9 = 261

6. 칠판에 연속된 일곱 개의 자연수가 적혀 있다. 이 중에 하나의 수를 지우고 나머지를 더했더니, 2015였다. 지운 수는 무엇인가?

① 331　　② 334　　③ 337　　④ 340　　⑤ 343

** 출제의도?
계산식 만들기
** 의미?
답을 얻어낼 수 있는 계산식을 만들고 답을 찾는다.
** 해결예시?
1) 주어진 조건에 따라 다음과 같은 연속된 일곱 개의 자연수를 가정한다.
(x-3, x-2, x-1, x, x+1, x+2, x+3)
2) 이 중 지운 숫자를 x+i라고 가정한다.
3) 따라서 7x-(x+i) = 2015라는 식이 성립한다.
4) 이것을 정리하면 6x-i=2015이며 i는 -3~3사이의 자연수여야 한다.
5) 2015를 6으로 나누면 몫이 335이고 나머지가 5이다.(6x335+5=2015)
6) 이것을 6x-i=2015와 같은 형태로 정리하면 6x336-1 이다.
7) 따라서 x는 336, i는 1이므로 지운 숫자는 337이다.

7. 어느 우물 안에는 개구리가 한 마리 살고 있다. 이 개구리는 우물의 바닥에서부터 벽을 타고 기어올라 우물 밖으로 나오려고 한다. 개구리는 낮 동안 우물 벽을 타고 3m를 기어오를 수 있지만 밤에는 잠을 자며 그동안 2m를 미끄러진다고 한다. 우물의 깊이가 정확히 19.4m라면 오늘 낮부터 시작하여 이 개구리가 우물 밖을 나오기 까지 최소 몇 밤을 우물 안에서 잠들어야 할까?

① 17 ② 18 ③ 19 ④ 20 ⑤ 21

** 출제의도?
관계식 만들기
** 의미?
점화식 만들기와 유사
** 해결예시?
개구리의 위치는 3 - 1 - 4 - 2 - 5 - 3 - … - 20 - 8 이 된다.
최초로 19.4m를 넘는 경우는 18일째(20 - 3 + 1) 이므로 최소 17밤을 잠들어야 한다.
1) 잠든 밤의 숫자를 i라고 가정하고 우물을 올라가는 최대 높이를 반환하는 f(i)라는 함수를 만든다.
2) 오늘 낮은 f(0)이고 f(0)의 값은 3, f(1)은 f(0)-2+3이므로 4이다.
3) 이 점화식을 정리하면 f(i) = i+3임을 알 수 있다.
4) 따라서 f(i)값이 19.4보다 크거나 같은 i의 최솟값은 17임을 알 수 있다.

8. 어떤 프로그램은 임의의 자연수에 대해서 그 결과가 1이 나올 때 까지, 아래의 연산을 반복한다.

그 자연수가 짝수라면 2로 나눈다. 홀수라면 3을 곱하고 1을 더한다.

예를 들어, 자연수 5에서 시작했다면, 5 -> 16 -> 8 -> 4 -> 2 -> 1의 순서로 값이 변하며 마지막에 1에 도달하게 된다. 이때에 실행한 연산의 횟수는 5회이다. 그렇다면 682는 위의 연산을 몇 회 실행했을 때 1에 도달하게 될까?

① 7 ② 10 ③ 12
④ 13 ⑤ 1에 도달하지 못함

** 출제의도?
계산과정 이해하고 따라가기(트레이싱)
** 의미?
콜라츠 추측, 우박수
** 해결예시?
콜라츠의 추측, 3n+1 문제, 우박수 문제
682 -> 341 - 1024 이다.
1024는 2^10이므로 연산을 12회 하면 1이 된다.

9. 키가 모두 다른 8명의 사람들 A, B, C, D, E, F, G, H가 어떤 순서대로 한 줄로 서 있다. 각자가 자신보다 앞에 서 있는 사람들 중 키가 작은 사람들의 수 x와, 자신보다 뒤에 서 있는 사람들 중 키가 작은 사람들 수 y를 세어 순서쌍 (x, y)로 표시하였더니 다음과 같았다.

A:(1, 1), B:(6, 1), C:(0, 3), D:(0, 0), E:(0, 4), F:(5, 0), G:(1, 0), H:(3, 3)

앞에서 4번째 사람은 누구인가?

① A ② B ③ C ④ E ⑤ H

** 출제의도?

데이터 해석

** 의미?

주어진 데이터를 해석해 원하는 답 찾아내기

** 해결예시?

1) 문제를 분석해보면 A는 3번째, B는 8번째, C는 4번째, D는 1번째, E는 5번째, F는 6번째, G는 2번째, H는 7번째로 크다는 것을 알 수 있다.

사람	A	B	C	D	E	F	G	H
순서쌍	(1,1)	(6,1)	(0,3)	(0,0)	(0,4)	(5,0)	(1,0)	(3,3)
키 순서	3	8	4	1	5	6	2	7
위치	?	?	?	?	?	?	?	?

2) 모든 정보를 갖고 있는 가장 작은 B부터 위치를 정하면 아래와 같은 정보를 알 수 있다.

사람	A	B	C	D	E	F	G	H
순서쌍	(1,1)	(6,1)	(0,3)	(0,0)	(0,4)	(5,0)	(1,0)	(3,3)
키 순서	3	8	4	1	5	6	2	7
위치	5	7	2	3	1	8	6	4

3) 위치 번호에 따라 순서를 다시 배열하면 아래와 같은 정보를 알 수 있다.

사람	E	C	D	H	A	G	B	F
순서쌍	(0,4)	(0,3)	(0,0)	(3,3)	(1,1)	(1,0)	(6,1)	(5,0)
키 순서	5	4	1	7	3	2	8	6
위치	1	2	3	4	5	6	7	8

따라서 4번째 사람은 H다.

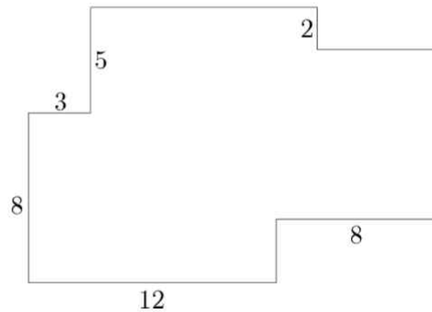
10. 아래처럼 네 자리 자연수 ABCD에 D를 곱했더니 네 자리 자연수 DCBA가 되었다. A, B, C, D는 모두 0 이상 9 이하의 서로 다른 숫자이다. D는 몇일까?

$$\begin{array}{r} A\ B\ C\ D \\ \times \quad\quad D \\ \hline D\ C\ B\ A \end{array}$$

① 2 ② 3 ③ 4 ④ 7 ⑤ 9

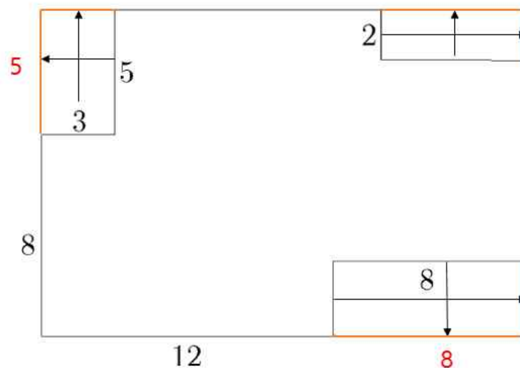
** 출제의도?
조건 찾아 활용하기
** 의미?
복면산
** 해결예시?
1) D에 올 수 있는 수는 제곱수가 자기 자신이 아닌 2, 3, 4, 7, 8, 9 이다.
2) D*D = A를 만족하면서 4자리를 유지하려면 A = 1, D = 9인 경우밖에 없다.
3) D가 9이므로 4자리를 유지하려면 B = 0인 경우밖에 없다.
4) A = 1, B = 0, D = 9를 이용하여 계산하면 C = 8임을 알 수 있다.
그러므로 D는 9이다.

11. 아래 그림은 둘레가 수평 혹은 수직인 선분으로 이루어진 어떤 도형에서 몇 개의 변의 길이를 표시한 것이다. 이 도형의 둘레의 길이는 얼마인가?



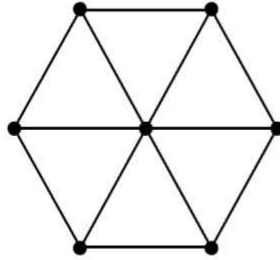
① 60 ② 62 ③ 64 ④ 66 ⑤ 68

** 출제의도?
도형의 이해
** 의미?
주어진 상황에서 미지 수 알아내기
** 해결예시?



도형이 수평 혹은 수직인 선분으로만 구성되어 있으므로 둘레 길이는 도형의 외곽을 직사각형 모양으로 확장했을 때의 둘레와 같다.
그러므로 20*2+13*2=66이 된다.

12. 아래 도형은 한붓그리기가 불가능하다. 즉, 종이에서 연필을 떼지 않고 모든 선분을 한 번씩만 지나도록 그리는 것은 불가능한 도형이다. 만약 같은 선분을 두 번 이상 지나는 것을 허용하여 연필을 종이에서 떼지 않고 한 번에 그린다면 두 번 이상 그려야 하는 선분의 최소 개수는 몇 개일까?



① 1 ② 2 ③ 3 ④ 4 ⑤ 5

** 출제의도?

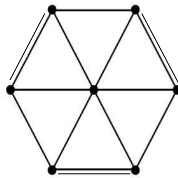
한 붓 그리기

** 의미?

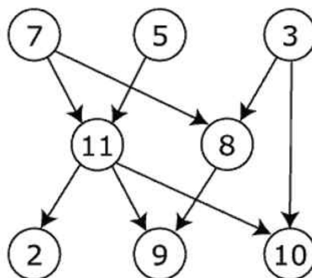
오일러 경로(홀 수 점의 개수 0 또는 2개) 만들기

** 해결예시?

각 정점에서 연결된 노드의 수를 구하고, 이 수를 변형함으로 한분 그리기가 가능한 도형으로 바꾼다. 이 때 추가된 노드가 중복해서 그리게 되는 선분이 된다.



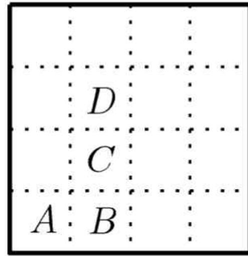
13. 다음은 여덟 종류의 작업을 수행하기 위해서 작업들 간의 순서 관계를 나타낸 도표이다. 예를 들어, 11번 작업을 수행하려면 7번과 5번 작업을 먼저 수행해야 한다. 보기의 순서대로 작업을 수행할 경우에 모든 작업이 정상적인 방식으로 완료될 수 없는 것은?

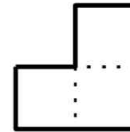


① 7 5 3 11 8 2 9 10
② 7 5 3 10 2 11 8 9
③ 5 7 3 8 11 10 9 2
④ 3 5 7 11 8 10 2 9
⑤ 5 3 7 11 8 2 9 10

** 출제의도?
작업 순서로 표현된 그래프 대한 이해
** 의미?
위상 정렬
** 해결예시?
작업들의 선후 관계를 생각해보면
7, 5, 3 그룹 -> 11, 8 그룹 -> 2, 9, 10 그룹 순으로 이루어져야 한다.
그룹 내에서의 순서는 관계가 없다.

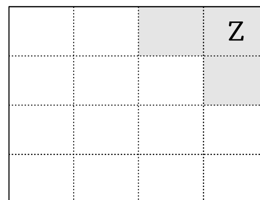
14. 명수네 집은 화장실 공사중이다. 명수네 집의 화장실은 아래의 왼쪽 그림과 같은 가로, 세로 4인 정사각형 모양이다. 화장실 바닥을 오른쪽 그림과 같은 모양의 타일로 채우려고 하는데, 화장실 바닥 중 한 칸은 하수도를 연결하기 위한 배수구로 사용해야 하기 때문에 타일로 채울 필요가 없다. 명수는 아래 그림에서 A, B, C, D 중 한 곳을 배수구의 위치로 하고 싶다. 화장실 배수구의 위치로 불가능한 곳은 어디일까?



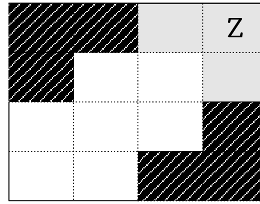


① A ② B ③ C ④ D ⑤ 없다

** 출제의도?
도형 채우기
** 의미?
논리적으로 도형 채워나가기
** 해결예시?
1) A,B,C,D를 제외한 모든 곳은 타일로 채워야하므로 Z는 반드시 채워져야 한다. 따라서 Z를 채우기 위한 모양은 아래의 음영을 채우는 방법 밖에 없다.



2) 같은 이유로 빗금친 부분도 반드시 채워야한다.



3) 따라서 남은 칸은 모두 7칸인데, 3개의 타일 2개를 채울 경우 한 칸이 남는다. 따라서 남은 칸에 모두 배수구를 위치시킬 수 있다.

15. 두 사람 A와 B는 4 이상의 자연수 N을 정한 후, 다음과 같은 게임을 한다.

(1) A는 1부터 시작하여 하나에서 세 개까지의 연속된 자연수를 외친다.
(2) B는 A가 부른 수의 다음 자연수부터 시작하여 하나에서 세 개까지의 연속된 자연수를 외친다.
(3) 다시 A는 B가 부른 수의 다음 자연수부터 시작하여 하나에서 세 개까지의 연속된 자연수를 외친다.
(4) (2)와 (3)을 반복하다 누구든지 N을 외치면 게임이 종료되어 승자가 된다.

예를 들면, N = 15일 때에, A가 외친 수들을 ( ) 안에, B가 외친 수들을 [ ] 안에 써서 표시하면 가능한 경우 중 하나는 (1, 2, 3) - [4, 5] - (6, 7, 8) - [9, 10, 11] - (12, 13, 14) - [15]로 B가 이기게 된다. A와 B가 N을 정하여 게임을 다시 시작하려 한다. 다음 중 A가 무슨 수를 써도 B가 항상 이기는 방법이 가능한 N은 무엇일까?

① 15 ② 16 ③ 17 ④ 18 ⑤ 없다.

** 출제의도?
논리적 추론
** 의미?
논리적 추론을 이용해 가능한 경우들 중에서 원하는 답을 찾아낸다.
** 해결예시?

[A]	[B]	[A]	[B]	[A]
1	2	3	4	5
2	3	4	5	6
3	**4**	5	6	7
	5	6	7	8
	6	**7**	8	9
		8	9	10
		9	**10**	11
			11	12
			12	13
				14
				15

A가 가능한 수는 홀수로 시작하고, B가 가능한 수는 짝수로 시작한다. 가능한 수의 끝은

모두 3의 배수로 끝난다. B가 반드시 이기려면 3N + 1의 수이면 된다.

16. C나 C++ 언어로 작성된 프로그램에서 여러 개의 구문을 한 덩어리로 묶을 때 사용하는 기호는?

① [와] ② { 와 } ③ < 와 > ④ “ 와 ” ⑤ \ 와 /

** 출제의도?
C언어 기초
** 의미?
코드 블록(중괄호)의 이해
** 해결예시?
…

17. 다음 프로그램이 출력하는 값은?

```
int q=0, s=1, t=1, i;
for(i=0; i<4; i++){
 q=s+t;
 t=s;
 s=q;
}
printf("%d", q);
```

① 2 ② 4 ③ 6 ④ 8 ⑤ 10

** 출제의도?
코드의 이해
** 의미?
피보나치 수
** 해결예시?
i=0 일 때, q = 2
i=1 일 때, q = 3
i=2 일 때, q = 5
i=3 일 때, q = 8 이므로 마지막에 출력되는 q의 값은 8이 된다.

18. 다음 중에서 변수의 이름으로 사용이 불가능한 것은?

① flood ② flown ③ flop ④ float ⑤ floof

** 출제의도?
C언어 기초
** 의미?

변수명 생성 규칙

** 해결예시?

- 영문, 숫자, 밑줄 문자(_)를 사용하며, 특수기호를 포함할 수 없다.(숫자 사용은 가능하나 숫자로 시작할 수 없으며, 숫자로만 변수명을 만들 수 없다.)
- C, C++언어에서 예약하고 있는 "예약어(keyword)"를 사용할 수 없다. 주요 예약어로 int, char, float, double, long, if, for, while, printf, return 등이 있다.
- 대문자와 소문자는 구별하여 사용한다.

19. 다음 중 C나 C++ 언어에서 정의된 기본 type이 아닌 것은?

① int ② float ③ char ④ typedef ⑤ long

** 출제의도?

C언어 기초

** 의미?

데이터형의 종류

** 해결예시?

이름	크기	저장할 수 있는 값의 범위
char	1byte	영문숫자 1문자 -128~127
unsigned char	1byte	영문숫자 1무자(부호없음) 0~255
short int	2byte	정수 -32768 ~32767
unsigned short int	2byte	정수(부호없음) 0 ~ 65535
int	4byte	정수 -2147483648 ~ 2147483647
unsigned int	4byte	정수(부호없음) 0 ~ 4294967295
long int	4byte	장정수 -2147483648 ~ 2147483647
unsinged long int	4byte	장정수(부호없음) 0 ~ 4294967295
float	4byte	단정밀도 부동소수점수 3.4E-38 ~ 3.4E+38
double	8byte	배정밀도 부동소수점수 1.7E-308 ~ 1.7E+308
long double	8byte	확장 배정밀도 부동소수점수 1.7E-308 ~ 1.7E+308

20. 다음 프로그램이 출력하는 값은?

```
int i, max, a[7]={1,8,9,10,10,2,6};
max=a[0];
for(i=1; i<7; i++){
  if(a[i]>max) max=a[i];
}
printf("%d", max);
```

① 6 ② 7 ③ 8 ④ 9 ⑤ 10

** 출제의도?

코드의 이해
** 의미?
배열의 최댓값 구하기
** 해결예시?
최댓값은?

21. 다음 프로그램의 출력 결과는 무엇인가?

```
int a=128, b=2015;
while(b!=0){
  int tmp=a%b;
  a=b;
  b=tmp;
}
printf("%d\n", a);
```

① 1 ② 2 ③ 3 ④ 4 ⑤ 5

** 출제의도?
코드의 이해
** 의미?
유클리드 호제법
** 해결예시?
128, 2015의 최대공약수는?

22. 정수형 변수 a의 값을 b의 값으로 나눈 나머지를 변수 c에 넣으려고 할 때 다음 빈 칸에 들어갈 연산자는?

```
c = a [    ] b;
```

① - ② + ③ * ④ / ⑤ %

** 출제의도?
C언어 기초
** 의미?
나머지 연산자
** 해결예시?
…

23. 다음 프로그램의 출력 결과는 무엇인가?

```
int arr[10]={0,1,1,1,0,0,1,1,1,1};
int cnt=0, i=0;
for(i=1; i<10; i++){
  if(arr[i]==arr[i-1]) cnt=cnt+1;
  else cnt=cnt-1;
}
printf("%d\n", cnt);
```

① 1 ② 2 ③ 3 ④ 4 ⑤ 5

** 출제의도?
코드의 이해
** 의미?
왼쪽 배열 값과의 비교
** 해결예시?
이전 수와 비교하여 같은 수이면 +1, 다른 수이면 -1을 한다. 같은 수 6개, 다른 수 3개이므로 출력 결과는 3이 된다.

24. 다음 프로그램의 출력 결과는 무엇인가?

```
int arr[10]={0,9,8,4,0,0,1,2,0,1};
int cnt=0, i;
for(i=0; i<9; i++) arr[i+1]+=arr[i];
printf("%d\n", arr[7]-arr[2]);
```

① 7 ② 9 ③ 5 ④ 1 ⑤ 3

** 출제의도?
코드의 이해
** 의미?
배열 값의 누적
** 해결예시?
이 코드는 arr배열의 값을 누적한 값으로 변경한다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
수행 전 arr	0	9	8	4	0	0	1	2	0	1
수행 후 arr	0	9	17	21	21	21	22	24	24	25

25. 다음 프로그램의 출력 결과는 무엇인가?

```
printf("%d %d\n", (int)strlen("KOI\0"), (int)strlen("KOI\n"));
```

① 3 3 ② 3 4 ③ 3 5 ④ 5 4 ⑤ 5 5

** 출제의도?
C 함수의 이해
** 의미?
문자 개수 리턴 함수 strlen( )
** 해결예시?
strlen()은 문자열의 길이를 리턴하는 함수이다. 이때, 문자열의 끝을 NULL('＼0')문자로 인식하게 된다. 그러므로 (int)strlen("KOI＼0") 부분에서는 3을, (int)strlen("KOI＼n")에서는 4를 리턴한다.

26. 다음 프로그램의 출력 결과는 무엇인가?

```
int a=50, b=30;
printf("%d\n", a=b);
```

① 50 ② 20 ③ 0 ④ 30 ⑤ 1

** 출제의도?
대입 연산자의 이해
** 의미?
대입 연산 실행 후 왼쪽 변수에 저장 되어있는 값 출력
** 해결예시?
C언어에서는 '='기호는 대입연산자로 사용된다. 우측에 있는 연산 또는 데이터를 좌측으로 대입하라는 의미이다. printf("%d\n", a=b); -> b를 a에 넣고, 그 값을 출력한다.
그러므로 출력값은 b의 값인, 30이 된다.

27. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[5]={-12,25,-36,23,-1}, i, s=0;
for(i=0; i<4; i++){
  if(i%2==0) s+=a[i];
  else s=a[i]-s;
}
printf("%d\n", s*a[i]);
```

① 0 ② -22 ③ 22 ④ -96 ⑤ 96

** 출제의도?
코드의 이해
** 의미?

배열, 반복문

** 해결예시?

0번부터 3번 인덱스의 배열에 대해 짝수 인덱스의 값은 s에 누적하고, 홀수 인덱스의 값은 s와의 차를 대입하면 된다. 출력시 s*a[i]를 출력하는데 반복문이 종료되었을 때 i의 값이 i<4를 만족하지 않아야 끝남에 유의해야 한다.

짝수 일 때, s=s+a[i], 홀수 일 때, s=a[i]-s 를 수행한다. (i<4)

	[0]	[1]	[2]	[3]	[4]
a	-12	25	-36	23	-1
s	~~-12~~	~~37~~	~~1~~	22	

28. 다음 프로그램의 출력 결과는 무엇인가?

```
int a=2015, b=0, i;
for(i=1; i<=a; i++){
  if(a%i==0) b++;
}
printf("%d\n", b);
```

① 0 ② 1 ③ 2 ④ 4 ⑤ 8

** 출제의도?

코드의 이해

** 의미?

약수의 개수 구하기

** 해결예시?

약수의 개수를 출력하는 코드이다. 2015의 약수는 1 5 13 31 65 155 403 2015인 8개이다.

29. tribo(8, 4)의 결과는 무엇인가?

```
int tribo(int n, int k){
 if(n<=k) return n;
 return tribo(n-1, k)+tribo(m-2, k)+tribo(n-3, k);
}
```

① 85 ② 57 ③ 54 ④ 68 ⑤ 0

** 출제의도?

코드의 이해

** 의미?

다중(3중) 재귀의 이해

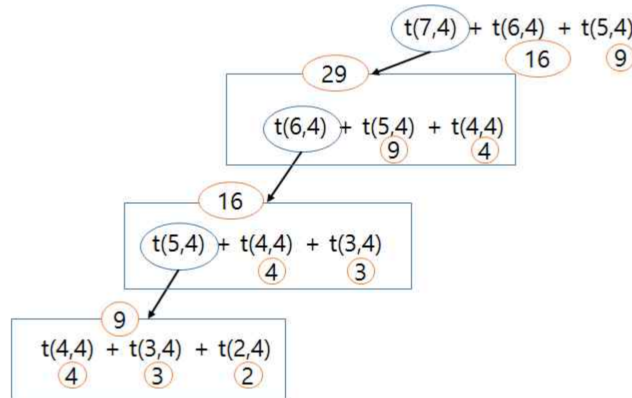
** 해결예시?

피보나치와 비슷하게 이전 세 수의 합이 다음 수를 만들어냄

. tribo(n, k) = tribo(n-1, k) + tribo(n-2, k) + tribo(n-3, k) 이므로

tribo(8, 4) = tribo(7, 4) + tribo(6, 4) + tribo(5, 4)가 된다. 조건문에 의해 n이 k이하이면 n값을 리턴하므로, tribo(1, 4) : 1, tribo(2, 4) : 2, tribo(3, 4) : 3, tribo(4, 4) : 4가 되므로 차례대로 계산하여 tribo(8, 4)를 구한다.

범위가 작은 경우 트리를 그려도 됨



30. 다음 프로그램의 실행 결과 sum의 값은 무엇인가?

```
int n=128, i, sum=0;
for(i=1; i<n; i++){
  if(n%i==0){
    sum+=i;
  }
}
```

① 127 ② 128 ③ 129 ④ 131 ⑤ 135

** 출제의도?

코드의 이해

** 의미?

약수의 합 구하기

** 해결예시?

128을 제외한 약수의 합을 출력하는 코드이다. 128의 약수는 1 2 4 8 16 32 64 이므로 이들의 합을 구하면 127이다.

31. 다음은 2차원 배열을 회전하여 출력하는 프로그램이다.

```
int arr[3][3]={
  {1,2,3},
  {4,5,6},
  {7,8,9}};
int i, j;
```

```
for(i=0; i<3; i++){
   for(j=0; j<3; j++){
     printf("%d ", [          ] );
   }
   printf("\n");
}
```

프로그램의 출력 결과가 다음과 같이 되기 위해 빈 칸에 들어갈 올바른 것을 고르시오.

```
7 4 1
8 5 2
9 6 3
```

① arr[i][2 - j]   ② arr[2 - i][j]   ③ arr[i][j]
④ arr[2 - j][i]   ⑤ arr[j][2 - i]

** 출제의도?
코드의 이해
** 의미?
배열의 활용
** 해결예시?
원래의 상태와 회전 후 상태를 통해 각 좌표의 계산식을 도출할 수 있다.

	[0]	[1]	[2]
[0]	1	2	3
[1]	4	5	6
[2]	7	8	9

가장 왼쪽의 화살표부터 순서대로 출력이 되므로, 배열에서 인덱스를 살펴보면
arr[2][0]   arr[1][0]   arr[0][0]
arr[2][1]   arr[1][1]   arr[0][1]
arr[2][2]   arr[1][2]   arr[0][2]   의 순서로 출력됨을 알 수 있다.

32. 다음 프로그램의 출력 결과는 무엇인가?

```
int A[5]={1,-2,0,3,5};
int i;
for(i=1; i<4; i++){
  printf("%d ", A[i]*2+A[i-1]+A[i+1]/4);
}
printf("\n");
```

① 0 0 2 ② 0 1 2 ③ 1 0 2
④ 1 1 2 ⑤ 0 0 3

** 출제의도?
코드의 이해
** 의미?
배열 참조
** 해결예시?
A[i] * 2 + A[i - 1] + A[i + 1] 부분을 보면 i번째 위치를 기준으로 왼쪽, 오른쪽 위치의 연속된 3개의 값을 참조해 계산이 이루어짐을 알 수 있다. for문에 의해 변화되는 i값을 이용해 계산을 하면 된다.

i=1 일 때, ((-2)*2+1+0)/4 = 0
i=2 일 때, (0*2+(-2)+3)/4 = 0
i=3 일 때, (3*2+0+5)/4 = 2

33. 다음 프로그램의 출력 결과는 무엇인가?

```
int A[100]={0,}, i, j;
for(i=1; i<100; i++){
  for(j=i*j; j<100; j+=i) A[j]++;
}
printf("%d\n", A[48]);
```

① 0 ② 1 ③ 2 ④ 4 ⑤ 5

** 출제의도?
코드의 이해
** 의미?
배열 채우기
** 해결예시?
1부터 99까지 숫자에 대해, i^2의 배수 위치마다 체크(누적)하므로 배열에 저장되는 값은 인덱스가 i^2의 배수에 해당하는 개수가 저장된다.
A[48]을 출력하므로 1부터 6까지 숫자에 대해 i^2의 배수에 해당하는 개수가 출력결과이다.
A[1] : $1(1^2)$의 배수에만 해당하므로 1이 저장
A[2] : $1(1^2)$의 배수에만 해당하므로 1이 저장
A[3] : $1(1^2)$의 배수에만 해당하므로 1이 저장

A[4] : $1(1^2)$의 배수, $4(2^2)$의 배수에 해당하므로 2가 저장
j가 48이 몇 번 되는지?
j는 i의 제곱에서 시작하므로, 시작이 i=7 이후에는 이미 48을 넘어서므로 계산할 필요가 없으며, 48은 1, 2, 3, 4, 6의 배수이므로 i가 5가지의 경우만 가능해 진다.

34. f(12)의 출력 결과는 무엇인가?

```
int f(int x){
  return (x/2.5)*3;
}
```

① 12 ② 13 ③ 14 ④ 15 ⑤ 18

** 출제의도?
C언어 기초
** 의미?
나누기 연산자, 형 변환의 이해
** 해결예시?
12/2.5*3 = (12/2.5)*3 = (4.8)*3 = 14.4 따라서 (int)(14.4) 정수형 변환시 버림 발생

35. 다음 프로그램의 출력 결과는 무엇인가?

```
char str[11]="abcdefghij";
int a=781030;
while(a>0){
  printf("%c", str[a%10];
  a/=10;
}
printf("\n");
```

① hibada ② hibadc ③ cdabih
④ adabih ⑤ hibaih

** 출제의도?
코드의 이해
** 의미?
배열 참조
** 해결예시?
변수 a를 10으로 나눈 나머지 값을 인덱스로 가지는 str의 데이터를 출력한다.
a=781030 이므로 출력순서는 str[0], str[3], str[0], stt[1], str[8], str[7]이다.

36. 다음 프로그램의 출력 결과는 무엇인가?

```
int A[4]={11,8,13,7};
int B[4]={2,6,4,10};
int i, j, ans=0;
for(i=0; i<4; i++){
  for(j=0; j<4; j++){
    if(A[i]+B[j]<=16 && A[i]+B[j]>ans) ans=A[i]+B[j];
  }
}
printf("%d\n", ans);
```

① 11　　② 8　　③ 15　　④ 13　　⑤ 25

** 출제의도?
코드의 이해
** 의미?
배열 참조, 최댓값 구하기
** 해결예시?
배열 A에서 하나의 수 와 배열 B에서 하나의 수를 선택하여, 그 두 수의 합으로 만든 수가 16이하 이면서, 가장 큰 수를 구한다.

37. 다음 프로그램은 두 자연수 a, b의 최대공약수를 구하는 함수이다. 빈 칸에 들어갈 코드는 무엇인가? (단, a <= b)

```
int gcd(int a, int b){
  int i;
  for(                    ){
    if(a%i==0 && b%i==0) return i;
  }
}
```

① i = 1; i <= a; i++　　② i = b; i >= a; i--
③ i = a; i >= 1; i++　　④ i = a; i >= 1; i--
⑤ i = a; i <= b; i++

** 출제의도?
코드의 이해
** 의미?
함수의 이해, 최대공약수 계산 구현
** 해결예시?
a<=b라는 단서가 주어져 있고, 최대공약수이므로 두 값 중 작은 a를 초과할 수 없다. 따라서 a부터 1씩 감소하면서 a, b 둘 다 나누어떨어지는 최대공약수를 찾도록 반복 구조를 작성해야 한다.

38. 다음 프로그램의 출력 결과는 무엇인가?

```
int p[7]={1,3,5,8,10,12,15};
int i, j, min=2015, ind;
for(i=1; i<=12; i++){
  int s=0;
  for(j=0; j<7; j++){
    if(p[j]>i) s+=p[j]-i;
    else s+=i-p[j];
  }
  if(min>s){
    min=s;
    ind=i;
  }
}
printf("%d\n", ind);
```

① 1　② 4　③ 8　④ 16　⑤ 26

** 출제의도?

코드의 이해

** 의미?

배열 참조, 누적값

** 해결예시?

배열값 / 수	1	3	5	8	10	12	15	누적값
1	0	2	4	7	9	11	14	47
2	1	1	3	6	8	10	13	42
3	2	0	2	5	7	9	12	37
4	3	1	1	4	6	8	11	34
5	4	2	0	3	5	7	10	31
6	5	3	1	2	4	6	9	30
7	6	4	2	1	3	5	8	29
8	7	5	3	0	2	4	7	28
9	8	6	4	1	1	3	6	29
10	9	7	5	2	0	2	5	30
11	10	8	6	3	1	1	4	33
12	11	9	7	4	2	0	3	36

39. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[10]={1,9,8,2,6,5,4,3,7}, i, j, c=0;
for(i=0; i<8; i++){
```

```
  for(j=0; j<8-i-1; j++){
    if(a[j]>a[j+1]){
      int temp=a[j];
      a[j]=a[j+1];
      a[j+1]=temp;
      c++;
    }
  }
}
printf("%d\n", c);
```

① 9　② 11　③ 13　④ 15　⑤ 17

** 출제의도?

코드의 이해

** 의미?

버블정렬의 데이터 교환 횟수 구하기

** 해결예시?

버블정렬 알고리즘에서의 데이터 교환횟수를 구하는 코드이다. 단, 위 코드에서 마지막 7은 빼고 정렬하게 되므로 17이 출력된다.

버블정렬로 오름차순 정렬을 진행하나 두 번째 for반복문을 보면 반복조건이 j < 8 - i - 1이다. i가 0일 때 반복조건이 'j값이 7보다 작을 동안'이 되고, a[j]와 a[j+1]을 비교하므로 배열의 8번 까지만 정렬이 됨에 유의해야 한다.

초기값	1, 9, 8, 2, 6, 5, 4, 3, 7
i=0	1, 8, 2, 6, 5, 4, 3, 9, 7
i=1	1, 2, 6, 5, 4, 3, 8, 9, 7
i=2	1, 2, 5, 4, 3, 6, 8, 9, 7
i=3	1, 2, 4, 3, 5, 6, 8, 9, 7
i=4	1, 2, 3, 4, 5, 6, 8, 9, 7
i=5	1, 2, 3, 4, 5, 6, 8, 9, 7
i=6	1, 2, 3, 4, 5, 6, 8, 9, 7
i=7	1, 2, 3, 4, 5, 6, 8, 9, 7

총 17회 데이터 교환 발생

40. f(0, 9, 5 6)의 값은 무엇인가?

```
int p[10]={1,3,7,8,4,3,5,2,9,2};
ing g(int a, int b, int x){
  int i, c=0;
  for(i=a; i<=b; i++) if(p[i]<=x) c++;
  return c;
}
int f(int a, int b, int x, int k){
  int mid, lc;
  if(a==b) return p[a];
  mid=(a+b)/2;
  lc=g(a, mid, x);
  if(lc>=k) return f(a, mid, x, k);
  else return f(mid+1, b, x, k-lc);
}
```

① 1 ② 2 ③ 3 ④ 5 ⑤ 8

** 출제의도?
코드의 이해
** 의미?
재귀 함수의 이해
** 해결예시?
lc는 g(a, mid, x)에 의해 결정된다. g()함수는 a부터 mid 사이에 있는 배열의 값 중 x이하 값의 수를 카운트하여 리턴 한다. lc와 k값 비교 결과, lc가 작으면 mid+1부터 b범위를, lc가 k이상이면 a부터 mid범위를 계속 탐색해 나간다.

재귀호출순서	f(0,9,5,6) => f(5,9,5,3) => f(8,9,5,1) => f(9,9,5,1)			
mid	4	6	8	p[9]
g(a, mid, x)	3	2	0	
if(lc>=k)	false	false	false	

41. 다음 프로그램의 출력 결과는 무엇인가? 단, 프로그램을 실행하는 컴퓨터는 ASCII 코드를 사용하는 것으로 가정한다.

```
printf("%c\n", 'B'+'Z'-'I');
```

① B ② R ③ S ④ T ⑤ Z

** 출제의도?
C언어 기초
** 의미?
ASCII 문자의 계산
** 해결예시?
아스키코드에서 알파벳(대문자 26개)은 연속적인 값을 가지고 있으므로, ‘A’를 1이라고 가

정했을 때, 'B'+'Z'-'I'는 2번째+26번째-9번째 = 19번째. 19번째 알파벳은 'S'이다.
※ 대문자 A의 아스키코드는 65이다.

42. 다음 프로그램의 출력 결과는 무엇인가?

```
int g(int x){
  if(x<10) return x;
  else return g(x/10)+(x%10);
}
int f(int a, int b){
  int c=1;
  while (b--) c*=a;
  return g(c);
}
int main(){
  printf("%d\n", f(2, 15));
  return 0;
}
```

① 26 ② 30 ③ 7 ④ 43 ⑤ 14783

** 출제의도?
함수의 이해
** 의미?
재귀함수의 이해
** 해결예시?
함수 f(a, b)는 a를 b번 곱한 값으로, f(2, 15)는 2^{15}(32,768)을 만들고, 함수 g(x)는 나누기와, 나머지를 이용해 각 자리수의 합을 구한다.
g(3276)+8 -> g(327)+6+8 -> g(32)+7+6+8 -> g(3)+2+7+6+8 -> 3+2+7+6+8 = 26

f()함수의 기능을 보면 b에 전달된 횟수만큼 a를 곱한 결과를 만들므로 g()함수에 전달되는 인수인 $c=a^b$임을 알 수 있다. g()함수의 g(x/10) + (x%10)을 리턴하는 부분을 보고 전달받은 인수를 어떻게 처리하고 있는지를 이해해야 문제의 답을 구할 수 있다.
2^{15} : 32768
g(32768) => g(3276) + 8 => g(327) + 6 + 8 ... => 출력 결과

43. 출력되는 결과가 0 1 2 3 4 5 6 7 8 이 되기 위하여 빈 칸 (a)에 들어가야 할 내용으로 알맞은 것은?

```
int a[5]={1,3,5,7}, b[6]={0,2,4,6,8}, c[10];
int i=0, j=0, n=4, m=5, k;
a[n]=100; b[m]=100;
for(k=0; k<n; k++) [          ]
for(k=0; k<n+m; k++) printf("%d ", c[k]);
printf("\n");
```

① c[k] = (a[i] < b[j]) ? a[i++] : b[j++];
② c[k] = (a[i] > b[j]) ? a[i++] : b[j++];
③ c[k] = (a[i] < b[j]) ? a[++i] : b[j++];
④ c[k] = (a[i] > b[j]) ? a[i++] : b[++j];
⑤ c[k] = (a[i] < b[j]) ? a[++i] : b[++j];

** 출제의도?
코드의 이해
** 의미?
병합정렬의 이해
** 해결예시?
병합정렬의 방법으로, 두 배열의 데이터 중 작은 값을 배열 C에 순서대로 넣는다.

44. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[3][4]={{2,0,1,5},{0,4,1,1},{1,4,0,0}};
void f(int b){
  int i;
  for(i=0; i<4; i++) a[b][i]=a[(b+1)%3][(i+1)%4];
}
int main()
{
  int i, tot=0;
  f(1);
  f(2);
  f(0);
  for(i=0; i<3; i++) tot=tot+a[i][(i+1)%4];
  printf("%d\n", tot);
  return 0;
}
```

① 2 ② 4 ③ 5 ④ 6 ⑤ 9

** 출제의도?

코드의 이해
** 의미?
2차원 배열의 참조
** 해결예시?
a[b][i] = a[(b + 1) % 3][(i + 1) % 4]; 코드를 통해 2차원 배열의 어떤 값을 이용해 어떤 위치의 값을 변경하는 것인지를 파악하면 문제를 해결할 수 있다. 이때, main()함수에서 f(1)=>f(2)=>f(0) 순서로 2차원 배열의 값이 변경되며, 변경된 배열의 값을 참조하여 값을 변경해야 함에 유의한다.

f(1) 수행 시,

	0	1	2	3
0	2	0	1	5
1	0	4	1	1
2	1	4	0	0

->

	0	1	2	3
0	2	0	1	5
1	4	0	0	1
2	1	4	0	0

f(2) 수행 시,

	0	1	2	3
0	2	0	1	5
1	4	0	0	1
2	1	4	0	0

->

	0	1	2	3
0	2	0	1	5
1	4	0	0	1
2	0	1	5	2

f(0) 수행 시,

	0	1	2	3
0	2	0	1	5
1	4	0	0	1
2	0	1	5	2

->

	0	1	2	3
0	0	0	1	4
1	4	0	0	1
2	0	1	5	2

	0	1	2	3
0	0	0	1	4
1	4	0	0	1
2	0	1	5	2

색칠된 부분의 합은 2이다.

45. 다음 프로그램의 출력 결과는 무엇인가?

```
int A[6]={0,3,0,2,3,0};
int B[3]={1,2,1};
int i,j;
for(i=1; i<5; i++){
  int c=0;
  for(j=0; j<3; j++) c+=A[i+j-1]*B[j];
  printf("%d", c);
}
printf("\n");
```

① 6857 ② 5678 ③ 8765

④ 8675 ⑤ 6578

** 출제의도?
코드의 이해
** 의미?
배열의 참조
** 해결예시?

```
i=1   0+A[0]*B[0] = 0
      0+A[1]*b[1] = 6
      6+A[2]*b[2] = 6
i=2   0+A[1]*B[0] = 3
      3+A[2]*b[1] = 3
      3+A[3]*b[2] = 5
i=3   0+A[2]*B[0] = 0
      0+A[3]*b[1] = 4
      4+A[4]*b[2] = 7
i=4   0+A[3]*B[0] = 2
      2+A[4]*b[1] = 8
      8+A[5]*b[2] = 8 출력 결과는 6578이다.
```

46. 다음 프로그램을 보시오.

```
int F(int *A, int n){
 int i, prev=0, cnt1=0;
 for(i=n-1; i>=0; i--){
   if(A[i]<prev){
      [          ]
      for (; i<n; i++) A[i]=(--cnt1>0);
      return 1;
    }
    if(A[i]==1) cnt1++;
    prev=A[i];
  }
  return 0;
}
int main(){
  int A[6]={0,0,0,1,1,1};
```

```
  do{
    for(int i=0; i<6; i++) printf("%d", A[i]);
    printf("\n");
  }while(F(A, 6));
  return 0;
}
```

위 프로그램을 실행한 결과 다음과 같은 출력이 나왔다. 빈 칸에 들어갈 알맞은 내용을 고르시오.

```
000111
001011
001101
001110
010110
011010
011100
101100
110100
111000
```

① A[i++] = 1;
   A[i++] = 0;
② A[i++] = 0;
   A[i++] = 1;
③ A[i] = 0;
④ A[i++] = 0;
⑤ A[i++] = 1;

** 출제의도?
코드의 이해
** 의미?
함수의 이해, 부분 문자열 바꾸기
** 해결예시?
배열의 끝(인덱스:5)에서부터 시작해 배열 데이터가 0일 때, 0인 값을 1로, 그 이전 값을 0으로 바꾼다.
000111에서 시작해서 111000으로 1을 앞으로 옮기는 작업?
오른쪽 끝에서부터 원하는 부분 문자열이 바꾸는 작업을 하는 프로그램이다. prev변수의 역할과 if(A[i]<prev) 조건문이 어떤 상황을 체크하고자 하는지, 그리고 출력된 문자열의 변화를 이해하면 어떤 문자열을 찾아 어떻게 바꾸고자 하는지를 이해할 수 있다.

000111
001011
001101
001110
010110
011010
011100
101100
110100

111000

47. 다음 프로그램의 출력 결과는 무엇인가?

```
int f(int x, int L){
  return 2*L-x;
}
int main(){
  int i, x=1;
  for(i=0; i<6; i++){
    x=f(x, -i);
    x=f(x, i);
  }
  printf("%d\n", x);
  return 0;
}
```

① 60 ② 61 ③ 62 ④ 63 ⑤ 64

** 출제의도?
코드의 이해
** 의미?
함수의 이해
** 해결예시?
f()함수를 한번 실행할 때마다 x의 절대값은 2*i만큼 증가함

48. 다음과 같은 함수 f가 있을 때, (a)와 (b)에 들어가는 값 중 다른 결과를 출력하는 문자열 쌍은 무엇인가?

```
int f(char* a, char* b){
  if((*a)=='\0' || (*b)=='\0') return 1;
  int c;
  if((*a)==(*b)) c=-1;
  else c=1;
  return c+f(a+1, b+1);
}
int main(){
  char A[]="  (a)  ";
  char B[]="  (b)  ";
  printf("%d\n", f(A, B));
  return 0;
}
```

① (a) abcd (b) abde
② (a) ABCD (b) abCD
③ (a) xx (b) xy
④ (a) KOI (b) IOI
⑤ (a) CODING (b) MODE

** 출제의도?
코드의 이해
** 의미?
함수의 이해, 포인터의 이해
** 해결예시?
f()함수에서 c값이 -1이 되는 경우와 1이 되는 경우가 언제인지가 파악되면 출력값을 찾을 수 있다.

[49-50] 다음과 같은 문제를 해결하기 위해 프로그램을 작성하였다. 물음에 답하시오.

어떤 공연장에서는 가로로 C개, 세로로 R개의 좌석이 C × R 격자형으로 배치되어 있다. 각 좌석의 번호는 해당 격자의 좌표 (x, y)로 표시된다.

예를 들어보자. 아래 그림은 가로 7개, 세로 6개 좌석으로 구성된 7×6 격자형 좌석배치를 보여주고 있다. 그림에서 각 단위 사각형은 개별 좌석을 나타내며, 그 안에 표시된 값 (x, y)는 해당 좌석의 번호를 나타낸다. 가장 왼쪽 아래의 좌석 번호는 (1, 1)이며, 가장 오른쪽 위 좌석의 번호는 (7, 6)이다.

(1, 6)						(7, 6)
			(4, 4)			(7, 4)
(1, 3)					(6, 3)	
(1, 2)						
(1, 1)	(2, 1)	(3, 1)				(7, 1)

이 공연장에 입장하기 위하여 많은 사람이 대기줄에 서있다. 기다리고 있는 사람들은 제일 앞에서부터 1, 2, 3, 4, ... 순으로 대기번호표를 받았다. 우리는 대기번호를 가진 사람들에 대하여 (1, 1) 위치 좌석부터 시작하여 시계방향으로 돌아가면서 비어있는 좌석에 관객을 순서대로 배정한다. 이것을 좀 더 구체적으로 설명하면 다음과 같다.

먼저 첫 번째 사람, 즉 대기번호 1인 사람은 자리 (1, 1)에 배정한다. 그 다음에는 위쪽 방향의 좌석으로 올라가면서 다음 사람들을 배정한다. 오른쪽에 더 이상 빈자리가 없으면 오른쪽으로 가면서 배정한다. 오른쪽에 더 이상 빈자리가 없으면 아래쪽으로 내려간다. 그리고 아래쪽에 더 이상 자리가 없으면 왼쪽으로 가면서 남은 빈 좌석을 배정한다. 이 후 왼쪽으로 더 이상의 빈 좌석이 없으면 다시 위쪽으로 배정하고, 이 과정을 모든 좌석이 배정될 때까지 반복한다.

아래 그림은 7×6 공연장에서 대기번호 1부터 42번까지의 관객이 좌석에 배정된 결과를 보여주고 있다.

6	7	8	9	10	11	12
5	26	27	28	29	30	13
4	25	38	39	40	31	14
3	24	37	42	41	32	15
2	23	36	35	34	33	16
1	22	21	20	19	18	17

여러분은 공연장의 크기를 나타내는 자연수 C와 R이 주어져 있을 때, 대기 순서가 K인 관객에게 배정될 좌석 번호 (x, y)를 찾는 프로그램을 작성해야 한다.

입력 형식

입력파일의 이름은 input.txt로 한다. 입력파일의 첫 줄에는 공연장의 격자 크기를 나타내는 정수 C와 R이 하나의 공백을 사이에 두고 차례대로 주어진다. 두 값의

범위는 5 ≤ C, R ≤ 1,000이다. 그 다음 줄에는 어떤 관객의 대기번호 K가 주어진다. 단, 1 ≤ K ≤ 100,000,000이다.

출력 형식

출력파일의 이름은 output.txt이다. 여러분은 입력파일에 제시된 대기번호 K인 관객에게 배정될 좌석번호 (x, y)를 구해서 두 값, x와 y를 하나의 공백을 사이에 두고 출력해야 한다. 만일 모든 좌석이 배정되어 해당 대기번호의 관객에게 좌석을 배정할 수 없는 경우에는 0(숫자 영)을 출력해야 한다.

입력과 출력의 예 1

입력

```
7 6
11
```

출력

```
6 6
```

입력과 출력의 예 2

입력

```
7 6
87
```

출력

```
0
```

입력과 출력의 예 3

입력

```
100 100
3000
```

출력

```
9 64
```

프로그램

```
#include <stdio.h>

int yy[]= (a) , xx[]= (b) ;
```

```
int T, R, C, A[1001][1001], Y[1000001], X[1000001];

int main(){
  int y, x, n, d;
  freopen("input.txt", "r", stdin);
  freopen("output.txt", "w", stdout);
  scanf("%d%d", &C, &R);
  for(n=x=1, y=d=0; n<=R*C; n++){
    int ny=y+yy[d], nx=x+xx[d];
    if(!(1<=ny && ny<=R && 1<=x && nx<=C && !A[ny][nx])){
      d=(d+1)%4;
      ny=y+yy[d], nx=x+xx[d];
    }
    y=ny, x=nx;
    A[y][x]=n;
    Y[n]=y, X[n]=x;
  }
  scanf("%d", &n);
  if(1<=n && n<=R*C) printf("%d %d\n", X[n], Y[n]);
  else puts("0");
}
```

49. 다음 중 (a)와 (b)에 들어갈 내용으로 알맞은 것은?

① (a) : {-1, 0, 1 0}, (b) : {0, -1, 0, 1}
② (a) : {-1, 0, 1, 0}, (b) : {0, 1 0, -1}
③ (a) : {1, 0, -1, 0}, (b) : {0, -1, 0, 1}
④ (a) : {1, 0, -1, 0}, (b) : {0, 1, 0, -1}
⑤ (a) : {0, -1, 0, 1}, (b) : {1, 0, -1, 0}

** 출제의도?
코드의 이해
** 의미?
배열의 참조 및 계산
** 해결예시?
공연장 좌석을 다음 좌석으로 바꾸기 위한 배열이므로 좌석을 배치하는 순서에 따라 배열을 채우면 된다.

50. 입력 파일 input.txt가 다음과 같이 주어졌을 때, 위 프로그램의 출력 결과는 무엇인가?

```
10 8
62
```

① 6 6　② 4 2　③ 5 3　④ 5 6　⑤ 7 5

** 출제의도?

코드의 이해

** 의미?

문제해결 프로그래밍의 이해

** 해결예시?

입력 데이터 샘플의 의미를 이해하고 해결하면 된다.

2015 시도예선 중고등부 문제

1. 1에서 10까지의 자연수를 모두 곱한 수를 x라고 하자. x를 8진수로 표기하면 제일 오른쪽에 연속으로 나타나는 0은 모두 몇 개일까?

① 1　　② 2　　③ 4　　④ 6　　⑤ 8

** 출제의도?

진법 변환

** 의미?

소인수 분해. 1~10의 자연수의 곱을 구하고 8진법으로 변환

** 해결예시?

1*2*3*4*5*6*7*8*9*10=3628800이다. 이 수를 8진법으로 변환하는 제일 오른쪽(1의 자리에서부터)의 0의 개수는 8로 나누어떨어지는 횟수를 의미한다.

각 수를 소인수 분해한 결과는 다음과 같다.

수	소인수분해 한 결과
1	1
2	**2**
3	3
4	**2 2**
5	5
6	**2** 3
7	7
8	**2 2 2**
9	3 3

2. 정점 6개가 있는 그래프에서, 각 정점마다 간선으로 연결된 다른 정점의 수를 세었더니 다음 보기와 같았다고 하자. 이 중 불가능 한 것은 무엇일까?

① 1, 1, 1, 2, 2, 2　　② 5, 5, 5, 5, 5, 5

③ 2, 2, 2, 2, 2, 2　　④ 3, 3, 4, 4, 4, 4

⑤ 0, 0, 0, 0, 0, 0

** 출제의도?

그래프 이론

** 의미?

그래프의 정점과 간선의 관계

** 해결예시?

6개의 정점이 있는 그래프의 각 정점에 대한 차수로 불가능 한 것을 고르는 문제이다.

일단 그래프에서 한 간선은 2개의 정점을 연결한다. 따라서 간선이 하나 추가될 때 마다 2개의 정점에 대한 차수가 1씩 증가한다. 전체적으로 볼 때 모든 정점의 차수의 합을 s라고 하고, 간선의 수를 e이라고 하자. 그러면 다음과 같은 식을 만족함을 알 수 있다.

$$s = 2e$$

즉, 항상 간선의 합은 짝수임을 알 수 있다. 따라서 주어진 각 정점들의 차수의 합을 구해보면 차수의 합이 홀수인 경우는 답이 될 수 없다.

3. 1 이상인 네 정수 a, b, c, d에 대해 a + b + c + d = 10을 만족하는 순서쌍 (a, b, c, d)의 가짓수는?

① 24 ② 84 ③ 112 ④ 120 ⑤ 216

** 출제의도?

경우의 수

** 의미?

조합. 가능한 경우의 수 찾기

** 해결예시?

1 이상인 네 정수 a, b, c, d의 합이 10이 되도록 하는 순서쌍 (a, b, c, d)의 가짓수 모든 수는 1 이상이므로 세 개의 수가 1인 경우에 나머지 한 수는 7이 될 수 있으며, 어떤 수도 7을 초과할 수 없다. 이런 원리를 이용하여 a가 1일 때 b는 1~7까지 가능하며, 또 b가 1일 때 c는 1~7까지 가능하고, a, b, c가 정해지면 d는 자연히 정해진다. 그러므로 a가 1, b가 1일 때 가능한 조합은 모두 7가지이다.

a가 여전히 1일 때 b가 2라면 c는 1~6까지 가능하여 a가 1, b가 2일 때 가능한 조합은 모두 6가지이다. 이런 식으로 따져보면 a가 1일 때 가능한 모든 조합은 7+6+5+4+3+2+1 = 28가지 조합이 나온다.

a가 2라면 b는 1~6까지 가능하고, b가 1일 때 c는 1~6까지 가능하며, a, b, c에 의해 d가 정해진다. a가 2, b가 1일 때 가능한 조합은 6가지, a가 2, b가 2일 때 가능한 조합은 5가지이다. 이런 식으로 따져보면 a가 2일 때 가능한 모든 조합은 6+5+4+3+2+1 이다. a가 1씩 증가할 때마다 가능한 조합의 가짓수는 1씩 감소한다.

a가 1일 때 : 7+6+...+2+1 가지

a가 2일 때 : 6+5+...+2+1 가지

a가 3일 때 : 5+4+...+2+1 가지

…

a가 6일 때 : 2+1 가지

a가 7일 때 : 1가지

따라서 모든 가짓수는 1*7 + 2*6 + 3*5 + 4*4 + 5*3 + 6*2 + 7*1 = 7+12+15+16+15+12+7 = 84가지

a	b	c	d	가짓수
1	1	1	7	7가지
		2	6	
		3	5	
		4	4	
		5	3	
		6	2	
		7	1	
	2	1	6	6가지
		2	5	
		3	4	
		4	3	
		5	2	
		6	1	
	3	1	5	5가지
		2	4	
		3	3	
		4	2	
		5	1	
	4	…	…	4가지
	5	…	…	3가지
	6	…	…	2가지
	7	…	…	1가지
…	…	…	…	…

다른 풀이

1이상인 네 자연수 $a,\ b,\ c,\ d$ 의 합이 10일 때, 순서쌍 $(a,\ b,\ c,\ d)$ 의 가짓수를 구하는 문제!!

모든 경우를 카운팅하여 구할 수도 있지만 시간이 많이 걸리므로 효율적으로 해결하기 위해서 2개의 순서쌍을 생각해보자.

1이상의 2개의 자연수의 합이 2부터 8까지인 경우의 수를 구하시오.

라는 부문제를 생각해보자. 두 자연수의 합이 10이 아니라 8인 이유는 남은 2개의 자연수의 합의 최솟값이 2이기 때문이다.

2자연수의 합이 k인 경우는 다음과 같이 쉽게 구할 수 있다.

$1+(k-1)$

$2+(k-2)$

:

$(k-1)+1$

로 모두 $k-1$가지라는 사실을 알 수 있다.

이 문제에 대한 해를 미리 구해두고 표로 정리해두면 보다 효율적으로 구할 수 있다.

합	2	3	4	5	6	7	8
가짓수	1	2	3	4	5	6	7

표 이름 dt

네 자연수의 합이 10인 경우는 두 자연수의 합이 2와 8인 경우, 3과 7인 경우, … , 8과 2인 경우뿐이므로 각 경우에 대해서 다음과 같이 구할 수 있다.

$$\sum_{i=2}^{8} dt[i]\,dt[10-i] = 7+12+15+16+15+12+7 = 84$$

즉 84가지임을 알 수 있다.

4. 5원, 8원 두 가지 종류의 우표가 있다. 이 두 종류의 우표를 이용하여 우편 요금을 지불하려고 한다. 우표의 개수에 제한이 없을 때 우편 요금이 어떤 값 x 이상이라면 항상 이 두 가지 우표를 이용하여 지불할 수 있다. x의 최솟값은 무엇인가?

① 24 ② 26 ③ 27 ④ 28 ⑤ 31

** 출제의도?

다이나믹?

** 의미?

표 채워나가기

** 해결예시?

처음에 5원과 8원을 지불할 수 있으므로 다음과 같이 체크해두고 시작하자.

5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
○			○																
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44

이제 x칸이 "○"라면 $x+5,\ x+8$ 은 모두 지불할 수 있는 금액임이 명확하므로 $x+5,\ x+8$ 칸에도 "○"표시를 하면서 x값을 증가시켜 나가면 된다.

그 결과는 다음과 같다.

5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
○			○		○			○		○	○		○			○			
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44

이와 같이 13까지 진행한 결과는 다음과 같다. 계속 더 진행해보자.

5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
○			○		○			○		○	○		○		○	○		○	○
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
○	○		○	○	○	○	○	○											

이 상태에서 확인해보면 28이후로 연속으로 5개의 칸이 모두 "○"로 채워져 있으므로 앞으로 모든 칸에 대해서 5원의 우표만으로도 "○"로 채워질 수 있음을 알 수 있다.

5. 아래처럼 네 자리 자연수 ABCD에 E를 곱했더니 네 자리 자연수 DCBA가 되었다. A, B, C, D, E는 모두 1 이상 9 이하의 서로 다른 숫자이다. D는 몇일까?

```
   A B C D
 ×       E
----------------
   D C B A
```

① 3 ② 4 ③ 7 ④ 8 ⑤ 9

** 출제의도?
특성 찾아 활용하기
** 의미?
복면산
** 해결예시?
A 와 E의 곱에서 올림이 발생하지 않는다. 따라서 A * E 는 9보다 작아야 한다. 혹시 3째 자리에서 올림이 발생하면 A * E 는 8보다 작아야 한다.

따라서 A와 E가 될 수 있는 숫자는(회색은 될 수 없음)

A	E	A*E = D	A*E+1 =D	판단
1	1	1	2	A와 E가 같으므로 될 수 없다.
	2	2	3	
	3	3	4	
	4	4	5	
	5	5	6	
	6	6	7	
	7	7	8	
	8	8	9	
	9	9	10	
2	1	2	3	
	2	4	5	A와 E가 같으므로 될 수 없다.
	3	6	7	
	4	8	9	
3	1	3	4	
	2	6	7	
	3	9	10	A와 E가 같으므로 될 수 없다.

D와 E의 곱은 A가 되어야 하므로 가능한 숫자는 (D, E, A) = (4, 8, 2) 뿐이다.
따라서 D는 4이다.

다른 풀이

이 문제는 복면산 문제로 다양한 패턴으로 자주 등장하는 문제이다. 모든 경우에 대해서 모두 시도해도 구할 수 있지만 주어진 정보를 이용하여 탐색의 범위를 줄이면 효율적으로

해결할 수 있다.

주어진 식은 다음과 같다.

$$ABCD \times E = DCBA$$

여기서 A, B, C, D는 모두 다른 자연수 이므로, E는 절대로 1은 될 수 없다.

따라서 E는 2부터 9까지의 값 중 하나이다. 그리고 $D \times E$ 의 1의 자릿수는 A, $A \times E$는 D라는 사실과 $A \times E$ < 10 이라는 사실을 이용하면 다음과 같이 구할 수 있다.

E가 2, 3일 때는 모든 값에 대해서 조사해도 위의 값을 만족시키는 값은 없다.

E가 2일 때 A의 값	1	3	4	5	6	7	8	9
$D=A \times E$	2	6	8					
$D \times E$	4	2	6					

E가 3일 때 A의 값	1	2	4	5	6	7	8	9
$D=A \times E$	3	6						
$D \times E$	9	8						

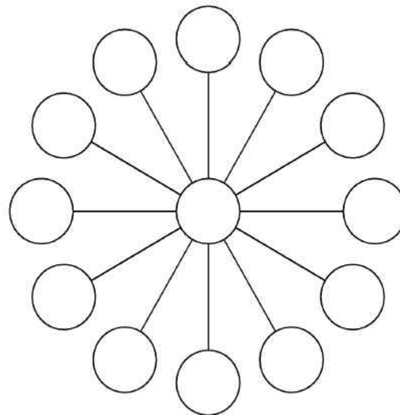
E가 4일 때에는 다음과 같이

E가 4일 때 A의 값	1	2	3	5	6	7	8	9
$D=A \times E$	4	8						
$D \times E$	6	2						

$A \times E = D \times E$ 인 값이 존재한다. 따라서 $A=2$, $E=4$, $D=8$ 일 경우 이 복면산의 식을 만족한다. **따라서 답은 8**

이와 같은 식으로 모두 구해보면 결과는 $2178 \times 4 = 8712$ 가 된다.

6. 1에서 13까지의 자연수를 아래 그림의 원 안에 하나씩 써넣을 때에 가운데 원을 지나는 직선 위에 놓인 세 숫자의 합이 모두 같도록 만들고 싶다. 가운데 원에 넣을 수 없는 수는 무엇인가?



① 1 ② 6 ③ 7 ④ 13 ⑤ 없다

** 출제의도?

패턴 만들기

** 의미?

규칙에 맞는 패턴을 찾아야 한다.

** 해결예시?

1부터 13까지의 수의 합은 91이다. 이 중 한 수를 가운데 원에 넣고 나머지 숫자들을 두 개씩 짝지었을 때 그 여섯 쌍의 값이 서로 같아야 한다. 만약 가운데 원에 1을 넣었다면 나머지들의 합은 91-1=90이다. 90을 6으로 나누면 15이다. 1을 제외한 나머지 숫자를 짝지어서 15(=2+13 = 3+12 = ... = 7+8)를 만들 수 있으므로 OK!

만약 가운데 원에 7을 넣었다면 나머지들의 합은 91-7=84이다. 84를 6으로 나누면 14이다. 7을 제외한 나머지 숫자를 짝지어서 14(=1+13 = 2+12 = ... 6+8)를 만들 수 있으므로 OK! 이렇게 따져보면 가운데 원에 13도 넣을 수 있다.

하지만 가운데 원에 6을 넣으면 나머지 숫자들의 합이 91-6=85이며 이것은 6으로 나누어지지 않는다. 따라서 원 안에 넣을 수 없는 수는 6이다.

다른 풀이

1부터 13까지의 자연수를 주어진 칸에 하나씩 써서 중점을 지나는 직선 상의 모든 수의 합이 일정하도록 만드는 것이 목적이다.

이 문제의 경우 가능한 모든 경우를 테스트하면 너무 많은 시간이 걸릴 수 있기 때문에 적절한 규칙을 찾아야 한다.

일단 이 문제에서 13개의 자연수 중에서 2개의 합이 같아지는 6개의 쌍을 찾고, 나머지 하나를 가운데 배치하는 전략으로 문제를 해결할 수 있다.

먼저 1부터 13까지의 자연수 중 1개를 제외하고 나열한 후 가장 오른쪽의 값과 왼쪽의 값을 묶으면 합이 같다.

1을 제외하고 가능한지 알아보자.

2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

위의 경우에는 (2, 13), (3, 12), ... , (7, 8)을 쌍으로 묶으면 모두 합이 15인 쌍으로 구성할 수 있다. 이 때 가운데 1을 쓰면 모두 16인 값을 만들 수 있다.

13을 제외할 경우에는

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

이 경우에도 (1, 12), (2, 11), ... , (6, 7)을 쌍으로 묶으면 모두 합이 13인 쌍으로 구성할 수 있다. 이 때 가운데 2를 쓰면 합이 26인 값을 만들 수 있다.

마지막으로 중간값인 7을 가운데 쓰면 (1, 13), (2, 12), ... , (6, 8)을 쌍으로 묶고 모두 합이 14인 쌍으로 구성할 수 있다. 이 때는 모두 합이 21인 값을 만들 수 있다.

이 외에는 다른 어떤 값을 쓰더라도 모두 합이 같이지도록 할 수 있는 방법은 없다. 따라서 답은 **보기 중 6이 불가능하다.**

다른 풀이

연속하는 숫자로 2등분하기 위해서는 아래와 같이 나누어야 한다.

1	2	3	4	5	6	7	8	9	10	11	12	13

세 숫자의 합은 16이다.

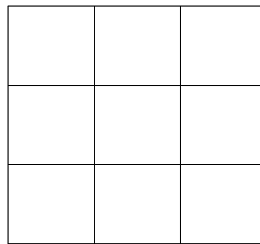
1	2	3	4	5	6	7	8	9	10	11	12	13

세 숫자의 합은 1+7+13=21이다.

1	2	3	4	5	6	7	8	9	10	11	12	13

세 숫자의 합은 1+12+13 = 26이다.

7. 1, 2, 3, 10, 11, 12, 19, 20, 21을 아래 그림과 같은 정방형 칸에 하나씩 넣어 가로, 세로, 대각선의 합이 모두 X로 같게 만들려고 한다. 가능한 X의 값은 무엇인가?



① 27 ② 29 ③ 31 ④ 33 ⑤ 35

** 출제의도?

패턴 만들기

** 의미?

마방진 만들기

** 해결예시?

1~9까지의 수로 정방형 칸에 넣을 때 가로 세로 대각선의 합은 1에서 9까지의 합 45를 3으로 나눈 값이다. 가로로 생각하면 3줄이 나오고 3줄의 가로의 합이 같아야 하기 때문에 45를 3으로 나눈 15의 값이 가로의 합이다.

8	1	6
3	5	7
4	9	2

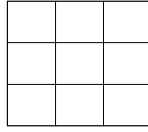
1부터 9까지의 수를 비교하여 1, 2, 3, 10, 11, 12, 19, 20, 21의 합은 99이다. 따라서 대각선의 합은 33이 된다.

20	1	12
3	11	19
10	21	2

다른 풀이

3행 3열의 칸에 주어진 수를 채워서 가로, 세로, 대각선의 합이 같아지도록 만드는 문제이다. 마방진으로 잘 알려진 문제를 조금 변형한 문제이다.

마방진 문제는 3행 3열의 칸이 있다면 1부터 9까지의 자연수를 채워서 가로, 세로, 대각선의 모든 합이 같아지도록 만드는 것이다.



하지만 이 문제는 1부터 9가 아니라 주어진 자연수의 집합 { 1, 2, 3, 10, 11, 12, 19, 20, 21 }을 이용하여 가로, 세로, 대각선이 같아지도록 만드는 문제이기 때문에 차이가 있다.

가능한 방법으로는 모든 경우의 수를 나열해보는 방법으로 도전할 수 있다. 이 경우에는 $9!$가지의 시도를 해야 하므로 주어진 시간 내에 해결하기 어렵다. 물론 적절히 가능한 부분을 고려하면 해결할 수도 있다.

이 문제는 실제로 시도할 필요가 없이 직관적으로 해결할 수 있는 방법이 있다.

이 문제에서 주어진 정보로 가로, 세로, 대각선의 합이 모두 같아야 하고, 문제의 해가 존재한다는 사실을 알고 있다.

따라서 9개 원소의 합을 구하고 이를 3으로 나눈 나머지가 결국 가로, 세로, 대각선의 합이 될 수밖에 없다.

왜냐하면 가로만 본다면 3개의 가로 합이 존재하고 이 값은 원소의 중복없이 9개의 모든 원소의 합과 같다.

따라서 전체 합을 3으로 나눈 값이 바로 원하는 해가 됨을 직관적으로 알 수 있다.

주어진 원소의 합을 구하면 99이다. 따라서 원하는 값은 33이 됨을 바로 구할 수 있다.

8. 어떤 자연수의 제곱으로 나타낼 수 있는 수를 제곱수라고 부른다. 예를 들어 1, 4, 9 등은 제곱수이다. 임의의 자연수는 여러 개의 제곱수의 합으로 표현할 수 있다. 예를 들어, 5 = 4 + 1, 7 = 4 + 1 + 1과 같이 쓸 수 있으며 5는 두 개의 제곱수의 합, 7은 4개의 제곱수의 합으로 나타낼 수 있다는 것을 알 수 있다. 그렇다면 89는 최소 몇 개의 제곱수의 합으로 표현할 수 있을까?

① 1 ② 2 ③ 3 ④ 4 ⑤ 5

** 출제의도?

경우의 수

** 의미?

제곱수의 합

** 해결예시?

89에 근접한 제곱수들을 모두 적어보자. 그리고 최소의 제곱수의 합으로 표현하기 위해서 제일 큰 제곱수부터 가능 여부를 따져본다. 81을 선택하면 다른 제곱수들로 7을 만들 수 없으므로 불가능하다. 64를 선택하면 25와 합하여 89를 만들 수 있으므로 합하여 89를 만들 수 있는 제곱수의 최소 개수는 2개이다.

1^2	2^2	3^2	4^2	5^2	6^2	7^2	8^2	9^2
1	4	9	16	25	36	49	64	81
				1			1	

다른 풀이

임의의 자연수를 제곱수의 합으로 나타내는 것에 관한 문제이다.

89를 5개 이하의 제곱수로 표현할 수 있을 때 최소로 몇 개의 제곱수로 표현하는지를 구하는 문제!!

먼저 89이하의 모든 제곱수를 나열하면 다음과 같다.

1, 4, 9, 16, 25, 36, 49, 64, 81

이 값을 이용하여 2개로 표현할 수 있는 모든 제곱수는 위의 값들을 적절히 조합하여 만들 수 있다.

81을 이용하여 만들 수 있는 자연수는 82, 85, 90 이므로 81을 이용하여 2개로 89는 만들 수 없다.

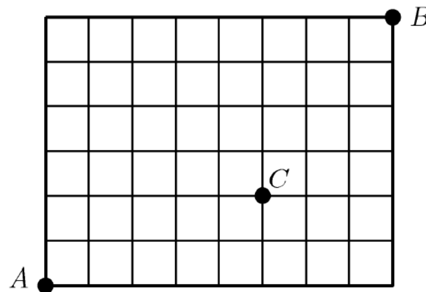
64를 이용하여 만들 수 있는 자연수는 65, 68, 73, 80, 89로 89는 65+25로 만들 수 있음을 알 수 있다.

이미 89를 구했으므로 더 이상 시도할 필요 없이 2개로 만들 수 있음을 알 수 있다.

따라서 답은 2개의 제곱수의 합으로 표현

이와 같은 방법을 너비우선탐색(BFS)라고 한다. 먼저 처음에 1개로 만들 수 있는 모든 해를 구한 후, 이 값들을 이용하여 2개로 만들 수 있는 방법을 모두 구하고, 마약 답이 없으면 이 답을 이용하여 3개, 4개 등으로 확장해 나가면 처음으로 찾는 값이 최소임이 명확하다.

9. 철수의 집은 A에 있다. 집에서 출발하여 B 지점에 있는 학교까지 걸어가야 하는데 중간에 C 지점에 있는 편의점을 들러서 가고 싶어 한다. 아래 그림은 A, B, C 지점의 주변을 약도로 표시한 것으로 각 선분은 철수가 지날 수 있는 길을 의미하며 가장 작은 정사각형의 가로, 세로 길이는 모두 정확히 1이다. 철수가 선택할 수 있는 최단 경로는 모두 몇 가지인가?



① 315 ② 420 ③ 525 ④ 735 ⑤ 1176

** 출제의도?

경우의 수

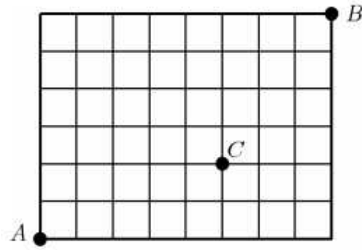
** 의미?

동적 프로그래밍(표 채워나가기)

** 해결예시?

이 문제는 A지점에서 출발하여 C를 반드시 거쳐서 B로 가는 최단경로의 수를 구하는 문제로 전형적인 동적프로그래밍 문제이다.

조합적인 아이디어를 이용해서도 풀 수 있지만 여기서는 표를 이용한 해결방법을 제시한다.



이와 같이 주어진 맵을 표로 바꾸자 이 때, 교차점을 칸으로 만들어야 하므로 7행 9열의 표가 필요하다. 표로 만들고 각 위치를 나타낸 결과는 다음과 같다.

								B
					C			
A								

위 표에서 각 칸에는 A로부터 출발하여 각 칸까지의 경우의 수를 쓴다. 단 최단경로로만 가므로 오른쪽과 위로만 이동할 수 있으며 C로 이동해야하기 때문에 따라서 처음에 다음과 같이 채울 수 있다.

								B
1					C			
1								
A	1	1	1	1	1			

임의의 어떤 칸으로 오는 경우의 수는 현재 칸 아래로부터 오는 방법과 왼쪽으로부터 오는 방법뿐이므로 아래 칸과 왼쪽 칸까지의 경우의 수를 이미 알고 있다면 두 값을 더하면 쉽게 구할 수 있다는 사실을 알 수 있다.

따라서 다음 음영으로 칠해진 값들을 구할 수 있다.

								B
1					C			
1								
A	1	1	1	1	1			

구한 결과는 다음과 같다.

								B
1	3	6	10	15	**21**			
1	2	3	4	5	6			
A	1	1	1	1	1			

이와 같이 C지점까지 21가지 서로 다른 방법이 있음을 알 수 있다. 다음으로 여기서 B까지 이동해야 하므로 다음과 같이 구성하여 음영으로 칠해진 부분의 값들을 같은 방법으로 구할 수 있다.

					21			B
					21			
					21			
					21			
1	3	6	10	15	**21**	21	21	21
1	2	3	4	5	6			
A	1	1	1	1	1			

음영 부분을 구해보면 다음과 같다.

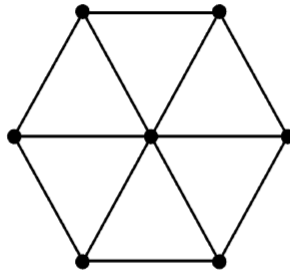
					21	105	315	**735**
					21	84	210	420
					21	63	126	210
					21	42	63	84
1	3	6	10	15	**21**	21	21	21
1	2	3	4	5	6			
A	1	1	1	1	1			

다른 풀이

A에서 C까지 갈 때 오른쪽으로 5번, 위로 2번 올라가야 한다. 총 7번의 이동에서 언제 위로 올라갈지 선택하면 된다. 따라서 ${}_7C_2$ 이다.

C에서 B까지 갈 때 오른쪽으로 3번, 위로 4번 올라가야 한다. 따라서 총 7번의 이동에서 언제 위로 올라갈지 선택하면 된다. 따라서 ${}_7C_3$ 이다.

10. 아래 도형은 각 꼭짓점 사이의 거리가 1인 도형이며, 한붓그리기가 불가능하다. 즉, 종이에서 연필을 떼지 않고 모든 선분을 한 번씩만 지나도록 그리는 것은 불가능한 도형이다. 만약 같은 선분을 두 번 이상 지나는 것을 허용하여 연필을 종이에서 떼지 않고 한 번에 그린다면, 연필의 이동거리는 최소 얼마가 될까?



① 12 ② 14 ③ 16 ④ 18 ⑤ 20

** 출제의도?

오일러 경로

** 의미?

오일러 한 붓 그리기 만들기

** 해결예시?

점에 만나는 선의 개수가 홀수인 점이 6개가 있다. 한붓그리기를 하려면 2개 이하가 되어야 한다. 따라서 선을 2개 더 그려 넣으면 홀수 인 점을 2개로 만들 수 있다.

총 모서리의 개수가 12개와 2개 더 그려 넣기 때문에 최소 이동거리는 14이다.

그래프에서 정점의 차수가 홀수인 점이 6개 존재하므로 한붓그리기가 불가능하다. 한붓그리기가 가능하기 위해서는 차수가 홀수인 점을 변으로 연결하는 것이다. 이렇게 되면 두 개의 점의 차수가 짝수로 바뀐다. 그리고 변을 주행할 때 이 변을 2번 이용하면 된다.

오일러회로를 만들기 위해서는 6개의 짝수점을 없애기 위해서 변을 3개 추가해야한다. 즉 모든 변의 수 + 3을 해야 모두 이동할 수 있다. 따라서 변의 수 12 + 5 = 15회만에 이동할 수 있다.

하지만 오일러 사이클을 만들기 위해서는 2개의 홀수점만 남기면 되므로 2개의 변만 추가하면 4개의 홀수점이 사라지고 2개의 홀수점만 남는다. 이 경우에는 변의 수 12 + 2 = 14이므로 14회 만에 모든 변을 이동할 수 있다.

다른 풀이

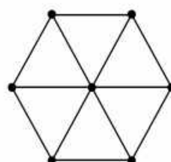
주어진 그래프가 한붓그리기가 불가능할 때, 주어진 그래프에서 변을 2번 이상 이용할 수 있도록 허용할 때, 모든 변을 지나기 위해서 최소로 이동해야할 거리를 구하는 문제이다.

기본적으로 그래프에서 한붓그리기가 가능하기 위해서는 오일러 경로 또는 오일러 회로의 조건을 만족해야 한다.

오일러경로가 존재하기 위해서는 그래프의 각 점의 차수들 중 홀수인 것이 2개 존재해야 한다. 이 경우 한 홀수점에서 다른 홀수점으로 오일러경로가 존재한다.

오일러회로가 존재하기 위해서는 그래프의 각 점의 차수들 중 홀수인 것이 없어야 한다. 이 경우에는 임의의 한 점에서 출발하여 그 점으로 도착하는 오일러사이클이 존재한다.

주어진 그래프는 다음과 같다.



이 그래프에서 정점의 차수가 홀수인 점이 6개 존재하므로 한붓그리기가 불가능하다. 한붓그리기가 가능하기 위해서는 차수가 홀수인 점을 변으로 연결하는 것이다. 이렇게 되면 두 개의 점의 차수가 짝수로 바뀐다. 그리고 변을 주행할 때 이 변을 2번 이용하면 된다.

오일러회로를 만들기 위해서는 6개의 짝수 점을 없애기 위해서 변을 3개 추가해야한다. 즉 모든 변의 수 + 3을 해야 모두 이동할 수 있다. 따라서 변의 수 12 + 5 = 15회 만에 이동할 수 있다.

하지만 오일러 사이클을 만들기 위해서는 2개의 홀수점만 남기면 되므로 2개의 변만 추가하면 4개의 홀수점이 사라지고 2개의 홀수점만 남는다. 이 경우에는 변의 수 12 + 2 = 14이므로 14회만에 모든 변을 이동할 수 있다.

11. 양의 정수들이 주어져 있을 때, 이들 중 정수 한 개 이상을 선택하고 이 선택한 정수를 모두 더하여 하나의 정수를 만들 수 있다. 예를 들어, 4개의 양의 정수 (1, 2, 1, 4)가 주어질 경우 다음과 같이 1부터 8까지의 모든 정수를 만들 수 있다 :

1, 2, 1+2=3, 4, 1+4=5, 2+4=6, 1+2+4=7, 1+2+1+4=8

10개의 양의 정수 (25, 10, 1, 1, 7, 2, 3, 104, 20, 30)가 주어질 때 여기에 어떤 양의 정수 하나를 더 추가하면 이들 11개의 정수들로부터 1부터 200까지의 모든 정수들을 만들 수 있다. 추가해야 할 정수의 최솟값은 무엇인가?

① 1 ② 2 ③ 3 ④ 4 ⑤ 5

** 출제의도?

경우의 수

** 의미?

조합 만들기

** 해결예시?

25, 10, 1, 1, 7, 2, 3, 104, 20, 30를 순서대로 정렬하면

1, 1, 2, 3, 7, 10, 20, 25, 30, 104가 된다.

작은 수를 하나씩 더해가면서 만들 수 있는 수의 범위를 계산하면 아래와 같다.

숫자	만들 수 있는 수의 범위
1	1
1, 1	1, 2(1+1)
1, 1, 2	1, 2, 3(1+2), 4(2+2)
1, 1, 2, 3	1~4, 5(2+3), 6(3+3), 7(4+3)
1, 1, 2, 3, 7	....
1, 1, 2, 3, 7, 10	....
1, 1, 2, 3, 7, 10, 20	....
1, 1, 2, 3, 7, 10, 20, 25	1 ~ 69
1, 1, 2, 3, 7, 10, 20, 25, 30	1 ~ 99
1, 1, 2, 3, 7, 10, 20, 25, 30, 104	100 ~ 103까지의 수는 만들 수 없음.

따라서 99에 어떤 수를 더해서 103까지의 수를 만들어야 하므로 4가 필요하다.

다른 풀이

주어진 양의 정수 10개에 적절히 1개의 자연수를 추가하여 이들의 조합으로 1부터 200까지의 자연수를 만들고자 한다.
이때 추가해야하는 적절한 자연수의 최솟값을 구하는 문제.
주어진 자연수는

25, 10, 1, 1, 7, 2, 3, 104, 20, 30

추가해야할 자연수의 최솟값은?
물론 모든 경우를 조사하는 방법으로 이 문제를 해결할 수는 없다.
보다 효율적으로 해결하기 위해서 자료를 오름차순으로 정렬하여 각 값들을 만들어 나가보자.

1, 1, 2, 3, 7, 10, 20, 25, 30, 104

이 집합을 S라고 하고 S_i를 S의 i번째 원소라고 하면 다음 두 원리가 성립함을 알 수 있다.

i) 만약 값이 $S_i - 1$인 원소가 존재하지 않는다면, $\sum_{k=1}^{i-1} S_k \geq S_i$ 를 만족해야 $S_i - 1$을 만들 수 있다.

ii) $S_i - 1$을 만들 수 있다면, 원소 S_i 이용하여 $2S_i - 1$까지 모든 자연수를 만들 수 있다.

이제 위 원리와 표를 이용해서 문제를 해결해보자.
먼저 104를 제외한 30까지의 표를 구성하고 각 S_i들을 “O”로 체크한다.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
○	○	○				○			○										○					○					○

다음으로 각 원소 S_i에 대해서 $2S_i - 1$까지는 채울 수 있다는 원리를 이용하여 가능한 칸에 모두 “▲”표시를 한다.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
○	○	○				○			○										○					○					○

먼저 이 표에서 3이 차 있으니 5까지는 모두 만들 수 있으므로 다음과 같이 채울 수 있다.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
○	○	○	▲	▲	▲	○			○										○					○					○

다음으로 7이 있으므로 15까지는 모두 가능하지만 일단 10이 있으므로 10이전까지만 채우자.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
○	○	○	▲	▲	▲	○	▲	▲	○										○					○					○

다음으로 10을 이용하여 19까지 모두 만들 수 있음을 알 수 있다.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
○	○	○	▲	▲	▲	○	▲	▲	○	▲	▲	▲	▲	▲	▲	▲	▲	▲	○					○					○

이와 같이 30이전의 모든 값은 다 채워짐을 알 수 있다.
이제 104이전 값이 103을 만들기 위해서 원리 1을 이용하여 모든 원소들의 합을 구해보자.

104미만의 모든 원소들의 합은 99임을 알 수 있다. 따라서 103을 만들기 위해서는 값이 4인 원소가 필요하다.

104까지 만들면 200까지 모든 원소를 만들 수 있다. 따라서 추가되어야 할 최소 원소의 크기는 4임을 알 수 있다.

12. 어떤 프로그램은 임의의 자연수에 대해서 그 결과가 1이 나올 때 까지, 아래의 연산을 반복한다.

그 자연수가 짝수라면 2로 나눈다. 홀수라면 3을 곱하고 1을 더한다.

예를 들어, 자연수 5에서 시작했다면, 5 -> 16 -> 8 -> 4 -> 2 -> 1의 순서로 값이 변하며 마지막에 1에 도달하게 된다. 이때에 실행한 연산의 횟수는 5회이다. 아래의 보기에 주어진 자연수 중 1이 되기까지 가장 많은 연산이 필요한 것은 무엇인가?

① 3 ② 5 ③ 7 ④ 8 ⑤ 9

** 출제의도?

문제의 분할

** 의미?

콜라츠 추측, 우박수

** 해결예시?

이 문제는 우박수라고 하는 유명한 문제이다. 이 문제는 특별한 규칙이 있는 것이 아니다. 규칙이 있다고 하면 2^k꼴의 수인 경우에 1까지의 길이가 k라는 점이 있다.

목적은 주어진 선택지 5개 중 가장 긴 단계를 필요로 하는 것을 구하는 것이 목적!!

특별한 규칙이 없으므로 각 보기별로 과정을 진행하는 방법으로 구할 수 있다.

이 때 그 과정에서 다른 보기의 수가 등장한다면 이 때 등장하는 수는 답이 되지 않음을 알 수 있다. 이 방법을 이용하여 효율적으로 해를 구할 수 있다.

먼저 선택지 ①인 3은 다음과 같다.

3 - 10 - 5 - 16 - 8 - 4 - 2 - 1

사실 16까지만 구해 봐도 답임을 알 수 있다.

그런데 위 과정에서 이미 5와 8이 등장했으므로 선택지 ②와 ④는 답이 되지 않음을 알 수 있다.

이제 선택지 ③인 7의 과정을 살펴보자.

7 - 22 - 11 - 34 - 17 - 52 - 26 - 13 - 40 - 20 - 10

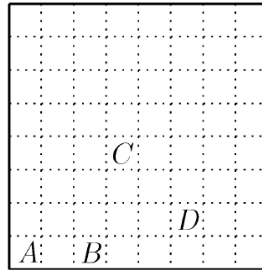
이 과정에서 10이 등장한다. 10은 이미 위에서 등장했기 때문에 더 이상해 볼 필요가 없다.

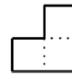
마지막으로 선택지 ⑤인 9의 과정을 살펴보자.

9 - 28 - 14 - 7

이 과정에서 7이 등장하므로 위의 과정을 반복하게 된다. 따라서 9가 가장 긴 과정을 거침을 알 수 있다.

13. 명수네 집은 화장실 공사중이다. 명수네 집의 화장실은 아래의 왼쪽 그림과 같은 가로, 세로 8인 정사각형 모양이다. 화장실 바닥을 오른쪽 그림과 같은 모양의 타일로 채우려고 하는데, 화장실 바닥 중 한 칸은 하수도를 연결하기 위한 배수구로 사용해야 하기 때문에 타일로 채울 필요가 없다. 명수는 아래 그림에서 A, B, C, D 중 한 곳을 배수구 위치로 하고 싶다. 화장실 배수구의 위치로 불가능한 곳은 어디일까?





① A ② B ③ C ④ D ⑤ 없다

** 출제의도?

문제의 분할

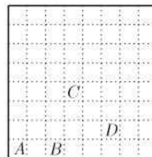
** 의미?

압축, 전체 탐색

** 해결예시?

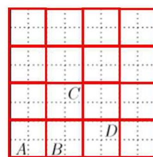
8행 8열의 격자에서 임의의 한 칸을 채우지 않으면서 "ㄱ"모양의 트리미노들을 이용하여 다른 모든 칸을 채우는 문제이다.

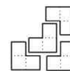
보기에 정말 복잡해 보이나 적절히 압축하여 해결하면 쉽게 해결할 수 있다. 주어진 맵은 다음과 같다.



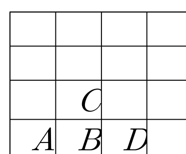


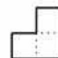
이 맵을 2행 2열을 하나의 칸으로 묶으면 다음과 같이 표현될 수 있다. 물론 "ㄱ"모양의 트리미노도 다음과 같이 압축한다.





그러면 문제를 다음과 같이 단순화 할 수 있다.





따라서 각 영역을 비우면서 채울 수 있느냐의 문제로 바뀐다. 만약에 압축된 그림에서 A영역을 비울 수 있다면 실제 맵에서 2행 2열의 정사각형에서 하나를 비우면서 채우는 문제로 바뀌기 때문에 당연히 비울 수 있다.
이와 같은 칸에서는 모든 영역을 비울 수 있음을 알 수 있다.
이를 다시 압축하면 다음과 같은 그림이 될 수 있음을 알 수 있다.

A,B,C	D

그리고 다시 4개의 트리미노로 하나의 큰 트리미노를 만들 수 있다. 이는 다음과 같이 채울 수 있다.

A,B,C	D

이와 같이 채우면 A, B, C를 남길 수 있으며 이는 다시 이전 단계의 압축으로 가면 다시 모두 남길 수 있음을 알 수 있다.
마찬가지로

A,B,C	D

로 채우면 D도 남길 수 있다.
사실 이 문제는 가로 세로의 크기가 2^k이라면 모든 원하는 칸을 비울 수 있다.

14. 10명의 사람들이 순서대로 줄을 서 있다. 앞에 있는 사람부터 차례대로 가지고 있는 구슬의 수가 10, 13, 26, 11, 15, 12, 18, 13, 25, 7 개이다. 사람들이 가지는 구슬의 개수를 모두 같게 하기 위하여 분배작업을 수행하려 한다. 분배작업 1회는 다음과 같다.

어떤 한 명이 인접한 한 사람에게 자신이 가지고 있는 구슬 중 일부를 준다.

모든 사람이 같은 구슬을 가지게 하는데 필요한 분배작업의 최소 횟수는 얼마인가?
① 5 ② 6 ③ 7 ④ 8 ⑤ 9

** 출제의도?
그리디(탐욕법)
** 의미?
그리디 알고리즘, 정렬
** 해결예시?
이 문제는 각 사람들이 모든 같은 구슬을 가지기 위해서 어떤 사람이 다른 사람에게 몇

개의 구슬을 건네줄 수 있다.

모든 사람이 같은 구슬을 가지기 위해서 건네주는 횟수를 최소화하는 것이 목적!!

처음 가지고 있는 구슬의 수는

10, 13, 26, 11, 15, 12, 18, 13, 25, 7

먼저 문제를 효율적으로 해결하기 위해서 오름차순으로 정렬해보자.

7, 10, 11, 12, 13, 13, 15, 18, 25, 26

그리고 10명이 가진 구슬의 합이 150개 이므로 각자 15개씩 가져야 한다.

따라서 15개 가진 사람은 구슬을 아무에게도 주지 않고 받지 않는 것이 이득임이 명백하므로 이 사람은 제외한다.

7, 10, 11, 12, 13, 13, 18, 25, 26

남은 사람들이 받아야 하는 사람은 +, 줘야하는 사람은 -로 다시 수를 고쳐보자.

+8, +5, +4, +3, +2, +2, -3, -10, -11

위 값에서 +3, -3은 1회의 분배로 처리할 수 있고 이보다 더 이득이 되는 방법은 없으므로 1회를 시행한다.

+8, +5, +4, +2, +2, -10, -11 (1회)

다음으로 +8, +2와 -10을 2회로 처리할 수 있으므로

+5, +4, +2, -11 (3회)

마지막으로 3회로 모두 분배할 수 있다. **따라서 답은 6회**

다른 풀이

10명의 평균은 15이다. 아래의 표는 필요한 구슬의 개수이다.

10	13	26	11	15	12	18	13	25	7
5	2	-11	4	0	3	-5	2	-10	8

10	13	26	11	15	12	18	13	25	7
5	2	-11	4	0	3	-3	2	-10	8
a	b	A	c	d	e	B	f	C	g

A가 a, b, c 에게 구슬을 나누어 주고 B가 e에게 C가 f와 g에게 구슬을 나누어 주면 된다.

A가 b에서 7개를 주고

b가 a에게 5개를 주고

A가 c에게 4개를 주고

B가 e에게 3개를 주고

C가 f에게 2개를 주고

C가 g에게 8개를 주면 끝난다. 따라서 총 횟수는 6회이다.

15. KOI 왕국의 통치자인 King Suryal은 마음씨가 좋다. 어느 날 King Suryal은 첩보를 통해 왕국 와인 저장고의 1000병의 와인 중에 한 병에 독이 들어 있다는 사실을 알아 내었다. 하지만 어떤 와인 병에 독이 든 것인지는 알 수가 없었다.

King Suryal은 부하들을 죽일 수는 없어서 생쥐 N마리에게 와인을 먹여서 독이 든 병을 찾기로 했다. 첩보에 의하면 독이 든 와인은 독이 너무나 강력하여 다른 와인과 섞어 얼마만큼 희석을 시킨다 하더라도 마신 후 정확히 30일 후에 무조건 사망한다고 한다. King Suryal은 오늘로부터 정확히 30일 후에 있을 축제에 와인을 사용할 예정이므로 각 생쥐에게 와인을 먹일 수 있는 횟수는 1회이다. 많은 생쥐를 사용하지 않으려 하므로 N을 최소화 했다고 한다. 이 N 마리의 생쥐 중 어떤 경우에라도 최대 K마리만을 희생시키려고 한다. 가능한 K의 최솟값은 얼마인가?

① 1 ② 8 ③ 9 ④ 10 ⑤ 999

** 출제의도?

퍼즐 해결

** 의미?

2진 체계 활용 퍼즐 해결

** 해결예시?

와인 병의 번호를 1번부터 1000번까지 매긴 뒤에 이 번호를 이진수 10자리로 나타낸다.(2^{10}=1024이므로 이진수 10자리로는 0부터 1023까지 표현할 수 있다.)

A, B, C, D, E, F, G, H, I, J라는 10마리의 쥐를 준비한다.

이진수의 자릿수 ↓

512	256	128	64	32	16	8	4	2	1	
0	0	0	0	0	0	0	0	0	1	1 ← 와인번호
0	0	0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	0	0	1	1	3
0	0	0	0	0	0	0	1	0	0	4
…	…	…	…	…	…	…	…	…	…	…
0	1	1	1	1	1	1	1	1	1	511*← 1이 9개 나타나는 번호
1	0	0	0	0	0	0	0	0	0	512
…	…	…	…	…	…	…	…	…	…	…
1	0	1	1	1	1	1	1	1	1	767*
…	…	…	…	…	…	…	…	…	…	…
1	1	0	1	1	1	1	1	1	1	895*
…	…	…	…	…	…	…	…	…	…	…
1	1	1	0	1	1	1	1	1	1	959*
…	…	…	…	…	…	…	…	…	…	…
1	1	1	1	0	1	1	1	1	1	991*
…	…	…	…	…	…	…	…	…	…	…
1	1	1	1	1	0	0	1	1	0	998
1	1	1	1	1	0	0	1	1	1	999
1	1	1	1	1	0	1	0	0	0	1000
A	B	C	D	E	F	G	H	I	J	← 쥐

J쥐에게는 2^0(=1)의 자리가 1인 와인을 조금씩 섞어 모두 먹인다.

I쥐에게는 2^1(=2)의 자리가 1인 와인을 조금씩 섞어 모두 먹인다.

…

A쥐에게는 2^9(=512)의 자리가 1인 와인을 조금씩 섞어 모두 먹인다.

그리고 30일 후에 죽은 쥐를 확인하면 몇 번 와인에 독이 있는지 알 수 있다. 죽은 것을 1로 표시, 산 것을 0으로 표시한다면 예를 들어, A, I, J 쥐만 죽었을 때, 이진수

1000000011, 즉 515번 와인이 독이 든 와인인 것이다. 그런데 이런 식으로 하면 최대 9마리까지 쥐가 죽을 수 있다. 0부터 1000 사이의 이진수에 1이 9번 들어가는 경우를 찾아보면 511, 767, 895 959, 991의 다섯 개이다. 죽는 쥐를 줄이기 위해 와인에 번호를 부여할 때 1이 9번 들어가는 이진수에 해당하는 번호는 건너뛰고 다음 번호를 부여한다. 예를 들어, 510번 와인 다음번 와인은 511번이 아니고 512번으로 부여하는 것이다. (1, 2, 3, ..., 510, 512, ..., 766, 768, ... 894, 896, ..., 958, 960, ..., 990, 992, ... 1000, ..., 1005) 이것이 가능한 이유는 이진수 10자리로 0부터 1023까지 표현할 수 있으나 현재 와인의 총 개수는 1000개이므로 여유 번호가 있기 때문이다. 따라서 10마리의 쥐를 사용하고 최대 8마리까지 희생시키면 독이 든 와인 병을 찾을 수 있다.

다른 풀이

일반적으로 정보과학적으로 볼 때, 1,000개의 자료를 이진화 하면 10개의 이진수가 필요하다.

따라서 10개의 컵을 준비한 후, 각 병을 0부터 1,000까지 혹은 1부터 999까지의 번호를 부여한 후, 이를 모두 이진수로 바꾼다.

각 이진수의 i번째 자리가 1이면 i번째 컵에 이 와인을 섞는다.

만약 10번 컵이라면 이를 10자리 이진수로 바꾸면

00 0000 1010

이 된다. 이와 같은 경우 이 병의 와인을 7번과 9번 컵에만 약간 섞으면 된다. 모든 와인에 대해서 이와 같은 작업을 한 후, 각 10마리의 생쥐들이 한 컵씩을 마신 후 한 달 뒤에 7번 쥐와 9번 쥐만 죽었다면, 10번병에 독이 들었다는 사실을 바로 알 수 있다.

그런데 여기서 재미있는 점은 이렇게 10개의 잔으로 표현할 수 있는 가짓수는 모두 1024가지라는 점이다.

실제 사용하는 값은 1,000개 뿐 이다. 그리고 모든 쥐가 죽기 위해서는 10자리에 모두 독을 타는 방법뿐이다. 이는

11 1111 1111

이 되는 경우인데 이 경우는 1,000 이상이 되어 이런 값은 절대 나오지 않는다. 즉, 10마리가 모두 죽게되는 경우는 없다. 따라서 최소 1마리는 항상 살 수 있다. 따라서 답은 9라고 생각할 수 있다.

9이하는 가능할까?

여기서 고민을 하게 되는데 코드화 방법에 따라서 적어도 24개의 값은 비울 수 있다. 그 이유는 실제 코드의 종류는 1024가지이고 사용하는 코드는 1000개이기 때문이다.

이때 10개의 잔 중 모든 잔에 타게 되는 경우의 수는 $_{10}C_{10}$가지가 된다. 즉 1가지이다.

다음으로 10개의 잔에 9잔에 타게 되는 경우의 수는 $_{10}C_{9}$가지이다. 이는 10가지이다.

즉 위 두 케이스를 모두 합해도 11가지뿐이다. 결론적으로 모든 1000병의 와인에 대해서 아무리 많은 잔에 섞는다고 해도 8잔을 초과하지 않으면서 섞을 수 있다. 왜냐하면 9개 이상의 잔에 섞어야만 하는 경우의 수는 위에서 소개한 11개뿐이므로 이는 24보다 작은 값이기 때문이다.

따라서 적절히 코드화를 잘하면 8마리만 희생하여 독이 든 와인병을 찾을 수 있다.

16. 다음 프로그램에서 출력되는 값은 얼마인가?

```
int a, i, j;
for(i=0; i<10; i++)
 for(j=0; j<5; j++)
   a=i+j;
printf("%d", a);
```

① 11　② 12　③ 13　④ 14　⑤ 15

** 출제의도?
코드 이해
** 의미?
중첩 반복 구조의 이해
** 해결예시?
변수 a의 최종값을 묻고 있으므로 반복문 안에서 가장 마지막으로 실행된 a의 상태를 찾으면 된다. 바깥 for문의 반복 조건이 위배되기 전 마지막 값이 i=9이고, 안쪽 for문의 반복 조건이 위배되기 전 마지막 값이 j=4이므로 a=9+4=13이 된다.

17. 다음 프로그램이 출력하는 값은?

```
int t, i;
t=0;
for(i=0; i<10; i++){
  t++;
}
printf("%d %d\n", t, i);
```

① 10 10　② 9 11　③ 10 9
④ 9 10　⑤ 11 9

** 출제의도?
코드 이해
** 의미?
반복 구조의 이해
** 해결예시?

i	0	1	2	3	4	5	6	7	8	9	10
t (초깃값 0)	1	2	3	4	5	6	7	8	9	10	10

i가 0부터 9까지 변하는 동안 for문의 블록이 반복 실행되면서 t값이 1씩 증가된다. i가 9일 때 i<10이라는 조건을 만족하므로 for문 안으로 진입하고 이 때 t값은 10이 된다. 그 다음에 i가 10이 되었을 때 i<10이라는 조건을 더 이상 만족하지 않으므로 for문으로 진입하지 않고 그 아래의 printf 문이 실행된다. 따라서 이 때 출력되는 i와 t는 둘 다 10이 된다.

18. 다음 중 변수의 이름으로 사용할 수 없는 것은?

① thisway ② int_char ③ star*star

④ that_way ⑤ _6_

** 출제의도?

C언어 기초

** 의미?

변수명 작성 규칙

** 해결예시?

- 중복된 변수명은 사용할 수 없다.
- 변수명은 영어 대소문자, 숫자, 밑줄(_)만 사용할 수 있다.
- 첫 글자로 숫자는 사용할 수 없다.
- C언어 키워드는 사용할 수 없다.

특수문자로는 유일하게 밑줄만 사용할 수 있으므로, 연산자 기호인 *는 변수명으로 사용할 수 없다.

19. 다음 중 정수형 변수의 type에 해당되는 것은 어느 것인가?

① class ② long ③ double

④ float ⑤ struct

** 출제의도?

C언어 기초

** 의미?

데이터 자료형

** 해결예시?

C언어의 기본 자료형은 크게 정수형, 실수형, 문자형이 있다. 그 중 정수형에 대한 문제이다. 그리고 정수형은 자료의 크기에 따라 다음과 같이 나눈다.

short : -32768~32767

int : -2147483648~2147483647

unsigned : 0~4294967295

long : -2147483648~2147483647

long long : -9223372036854775808~9223372036854775807

보기에 제시된 것 중 정수형 변수에 해당되는 것은 long 밖에 없다.

20. 다음 중에서 C나 C++ 언어의 키워드에 속하는 것은?

① main ② function ③ double

④ the ⑤ face

** 출제의도?

C언어 기초

** 의미?
예약어(키워드)
** 해결예시?
C언어에서 사용하는 기본 명령들이 있는데 이를 키워드라고 한다. 이 키워드는 기능을 가지고 있으므로, 변수명으로 사용할 수 없다. 이 문제에서 혼동하지 말아야 할 것은 변수명이나 함수명은 키워드에 포함되지 않아야 한다는 점이다. main 함수는 반드시 있어야 되는 함수지만 키워드가 아니다.

21. 다음의 코드를 실행하고 난 후 변수 a와 b에 들어가는 값을 순서대로 표시한 것은?

```
int a, b;
a=(21/4)*3;
b=(a*a)/a;
```

① 15, 15 ② 16, 15 ③ 16, 16
④ 15, 16 ⑤ 14, 15

** 출제의도?
C언어 기초
** 의미?
나눗셈 연산자의 정수 나누기
** 해결예시?
a = (21 / 4) * 3 ⇒ 5 * 3 = 15 (나눗셈 연산은 몫만 취하고 나머지는 버린다.)
b = (a * a) / a ⇒ (15 * 15) / 15 = 15
따라서 a, b는 15, 15

22. C나 C++ 언어로 작성된 프로그램에서 변수에 저장된 값을 비교할 때 사용하는 연산자가 아닌 것은?

① < ② <= ③ = ④ == ⑤ >=

** 출제의도?
C언어 기초
** 의미?
비교 연산자의 종류
** 해결예시?
코딩할 때 간혹 실수하는 문제이다. =과 ==은 의미가 완전히 다르다.
값을 비교할 때 사용하는 비교 연산자는 <, <=, ==, !=, >=, > 이다.
C언어에 =은 대입 연산자로서 변수에 값을 집어 넣을 때 사용하는 연산자이다.
(예: a=9 → a에 9를 대입)
==는 좌측과 우측의 값이 같은지 비교하는 연산자이다.(예: a==9 → a와 9가 같은지..)
비교 연산자의 연산 결과는 참(true, 1) 또는 거짓(false, 0) 이다.

23. 다음 중 scanf 함수에서 정수 값을 입력 받기 위해 사용하는 입력 형식은?

① “%d" ② "\r" ③ "%f" ④ "\n" ⑤ "%%“

** 출제의도?

C언어 기초

** 의미?

형식지정자

** 해결예시?

scanf(formatted scan 함수)에서 값을 입력 받기 위해 사용하는 형식 지정 문자에 대한 내용이다. 정수형 값을 입력받는 형식 지정문자는 %d이다.

형식 지정자는 키보드로 입력된 형태를 어떻게 인식해 저장할 것인지를 지정하는 것이라고 볼 수 있다.

24. 다음 프로그램의 출력이 “2 5 10 17 26”이 되도록 [ ] 부분에 들어갈 코드는 무엇인가?

```
int i;
i=1;
while(i<=5){
  printf("%d ", [          ]);
  i++;
}
printf("\n");
```

① i * i ② i + (i - 1) ③ i * (i + 1)

④ i / i + 1 ⑤ i * i + 1

** 출제의도?

코드의 이해

** 의미?

패턴 분석, 계산식의 유추

** 해결예시?

i	1	2	3	4	5
출력값	2	5	10	17	26

i와 출력값과의 관계를 잘 살펴보면 출력값은 i * i + 1 이라는 것을 알 수 있다.

25. 다음 프로그램의 출력 결과는 무엇인가?

```
char s[11]="hell world";
s[4]='\0';
printf("S%sE\n", s);
```

① ShellE　② Shell worldE　③ Shell E
④ Shell wolE　⑤ hello

** 출제의도?
코드의 이해
** 의미?
형식 출력(형식에 맞추어 출력)
** 해결예시?

s의 초기 상태	h	e	l	l		w	o	r	l	d	\0
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
s[4]='\0' 실행	h	e	l	l	\0	w	o	r	l	d	\0
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]

printf("S%sE\n", s); ← %s 위치에 s 배열의 문자열이 들어가는데 \0를 문자열의 끝으로 인식하므로 출력 결과는 ShellE 이 된다. %s는 문자배열에 들어가 있는 문자들을 널문자 \0를 만날 때까지 하나씩 모두 출력해 준다.

26. 다음 프로그램의 출력은 무엇인가?

```
int t, i;
t=0;
for(i=1; i<=10; i++) t+=i;
printf("%d\n", t);
```

① 45　② 50　③ 52　④ 53　⑤ 55

** 출제의도?
코드의 이해
** 의미?
1~n까지의 합
** 해결예시?
for문 변수 i는 1~10까지 1씩 증가하고 있고, 변수에 i값을 누적하고 있으므로 이 프로그램은 "1부터 10까지 합을 구하는 프로그램" 이다. 따라서 10*11/2=55 이다.

27. 다음 프로그램의 출력 결과는 무엇인가?

```
#define sq(x) x*x;
int main(){
  int z=4;
  printf("%d %d\n", sq(z), sq(z+1));
```

```
  return 0;
}
```

① 16 25 ② 16 9 ③ 16 16 ④ 25 25 ⑤ ERROR

** 출제의도?
  코드의 이해
** 의미?
  전처리 #define 의 이해
** 해결예시?
  #define 명령으로 정의된 매크로들은 컴파일되기 전에 내용을 치환하게 된다.
  이 문제의 printf 코드 부분을 치환한 코드는 다음과 같다.
  printf("%d %d\n", z*z, z+1*z+1); z=4이므로, 첫 번째 출력인 z*z는 16이 되고,
  두 번째 출력은 연산자 우선순위에 의해 1*z가 먼저 수행되므로 결국 z+z+1이므로 9가 된다.
  sq(z) → z * z = 4 * 4 = 16
  sq(z+1) → z + 1 * z + 1 = 4 + 1 * 4 + 1 = 9

28. 다음 프로그램의 출력 결과는 무엇인가?

```
int arr[10]={0,9,8,4,0,0,1,2,0,1}, cnt=0, i;
for(i=0; i<9; i++) arr[i+1]+=arr[i];
printf("%d\n", arr[7]-arr[2]);
```

① 3 ② 5 ③ 7 ④ 13 ⑤ 15

** 출제의도?
  코드의 이해
** 의미?
  배열 채워나가기
** 해결예시?

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
arr의 초기 상태 →	**0**	**9**	**8**	**4**	**0**	**0**	**1**	**2**	**0**	**1**
i=0	0	9								
i=1		9	17							
i=2			17	21						
…				…	…					
i=8										

  배열의 i번째에 저장되어 있는 값을 i+1번째에 누적하는 동작을 i가 0일 때부터 8이 될 때까지 반복 수행하고 나면 arr[7]은 arr[0]부터 arr[7]까지의 합과 같고, arr[2]는 arr[0]부터 arr[2]까지의 합과 같게 된다. 따라서 arr[7]-arr[2]의 계산 결과는 arr[3]부터 arr[7]까지의 합인 4+0+0+1+2=7이다.

29. 다음 프로그램의 출력 결과는 무엇인가?

```
int arr[10]={0,1,1,1,0,0,1,1,1,1};
int cnt=0, i=0;
for(i=1; i<10; i++){
  if(arr[i]==arr[i-1]) cnt=cnt+1;
  else cnt=cnt-1;
}
printf("%d\n", cnt);
```

① 1 ② 2 ③ 3 ④ 4 ⑤ 5

** 출제의도?
 코드의 이해
** 의미?
 이전 배열 값의 참조
** 해결예시?

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
arr의 초기 상태 →	0	1	1	1	0	0	1	1	1	1
cnt의 변화 →		-1	0	1	0	1	0	1	2	3

 i가 1부터 9까지 변하는 동안 i번째 배열의 값이 i-1번째 배열의 값과 같으면 cnt 값을 증가시키고, 그렇지 않으면 cnt 값을 감소시킨다. 최종적인 cnt의 값은 3이다.

30. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, b=0, c=0;
int a[7]={-3,-5,-6,-2,-1,-9,-4};
for(i=0; i<7; i++){
  if(b>a[i]) b=a[i];
  if(c<a[i]) c=a[i];
}
printf("%d\n", c=b);
```

① 7 ② 8 ③ 9 ④ 10 ⑤ 0

** 출제의도?
 코드의 이해
** 의미?
 최댓값, 최솟값 구하기
** 해결예시?
 반복문 안에 최솟값을 구하는 부분(b)과 최댓값을 구하는 부분(c)이 있다.
 여기서 주의해야 할 것은 b와 c의 초깃값이 0인데, 비교하는 배열 a의 값들이 음수라는 사실이다.
 최댓값 갱신이 한 번도 이루어지지 않으므로 c=0이 되고, b는 최솟값인 -9가 된다.
 따라서 c - 9 = 0 - (-9) =9이다.

31. 다음 프로그램의 출력 결과는 무엇인가?

```
int a=1, b=0, n=10;
while(n--){
  int c=b-a;
  a=b;
  b=c;
}
printf("%d\n", a+b);
```

① -2 ② -1 ③ 0 ④ 1 ⑤ 2

** 출제의도?
코드의 이해
** 의미?
과정 트레이싱
** 해결예시?
코드를 이해하고 n을 1씩 감소하면서 명령문을 따라가면 된다. 주의해야할 것은 n--는 후위 연산이므로 n의 상태가 0이 될 때까지 반복하는 것으로 생각하고 계산해야 한다.

보기값 / 반복회차	a	b	c
	1	0	
n=10	0	-1	-1
9	-1	-1	-1
8	-1	0	0
7	0	1	1
6	1	1	1
5	1	0	0
4	0	-1	-1
3			
2			
1	0	1	1

32. 다음 보기를 빈 칸에 넣었을 때 결과가 0인 것은 무엇인가?

```
int a[5]={ [          ] };
int i, s=0;
for(i=0; i<5; i++) s^=a[i];
printf("%d\n", s);
```

① 0, 1, 2, 3, 4 ② 1, 2, 3, 4, 4 ③ 2, 3, 4, 5, 6
④ 10, 20, 30, 40, 50 ⑤ 1, 1, 1, 1, 1

** 출제의도?

코드의 이해
** 의미?
비트단위 연산자 ^(XOR) 의 이해
** 해결예시?
^연산자는 비트 단위 XOR 연산자이며 두 개의 비트가 서로 다를 때 1을 반환한다.
s^=a[i]; ← s에 저장된 값과 배열의 i 번째 원소에 대해 비트 단위 XOR 연산을 수행하여(십진수를 이진수로 변환한 다음 같은 자리의 비트끼리 연산을 수행한다.) 그 결과를 다시 s에 저장하는 것을 배열의 처음부터 끝까지 반복 수행하였을 때 결과가 0이 되는 것을 찾는다. 어려운 작업이 아니므로 모두 수행해 본다. (아래 그림 참고) 따라서 s의 값이 0이 되는 a 배열의 원소들은 {1,2,3,4,4}이다.

①	0	1	2	3	4
a[i]	000	001	010	011	100
s	000	001	011	011	111

②	1	2	3	4	4
a[i]	001	010	011	100	100
s→	001	011	000	100	000

③	2	3	4	5	6
a[i]	010	011	100	101	110
s→	010	001	101	000	110

④	10	20	30	40	50
a[i]	001010	010100	011110	101000	110010
s→	001010	011110	000000	101000	011010

⑤	1	1	1	1	1
a[i]	001	001	001	001	001
s→	001	000	001	000	001

33. f(1, 16)의 값은 무엇인가?

```
int f(int a, int b){
 int mid;
 if(a==b) return 1;
 else{
  mid=(a*2+b)/3;
  return f(a, mid)+f(mid+1, b)+1;
 }
}
```

① 15　　② 16　　③ 31　　④ 32　　⑤ 33

** 출제의도?
코드의 이해
** 의미?
재귀 함수의 이해, 2진 탐색 변형
** 해결예시?
f(a, b) 함수는 a와 b의 값이 서로 같아질 때까지 중간값(m=(a*2+b)/3)을 기준으로 분리된다.
f(n, n) 형태일 때 1을 반환하고, f(n, n+1) 형태일 때는 f(n, n)+f(n+1, n+1)+1을 반환하므로 결국 3을 반환하게 된다. 또한 f(n, n+3) 형태일 때는 f(n, n+1)+f(n+1, n+2)+1을 반

환하므로 위의 f(n, n+1) 형태의 결과를 이용하면 7을 반환한다는 것을 알 수 있다. 이 정도의 규칙을 이용하여 문제를 풀어본다.

```
f(1, 16) = f(1, 6)                  + f(7, 16)                    + 1   ( m=6 )
         = f(1, 2) + f(3, 6) + 1 + f(7, 10) + f(11, 16) +1 + 1   ( m1=2,   m2=10 )
         = 3 + 7 + 1 + 7 + f(11, 12) + f(13, 16) + 1    + 2      ( m=12 )
         = 18 + 3 + 7 + 3
         = 31
```

다른 풀이

아래의 표와 같이 재귀함수로 나타낼 수 있다.

<table>
<tr><td colspan="8">f(1, 16)</td></tr>
<tr><td colspan="8">mid = (1 * 2 + 16) / 3 = 6</td></tr>
<tr><td colspan="8">f(1,16) = f(1,6) + f(7,16) + 1</td></tr>
<tr><td colspan="4">f(1,6)</td><td colspan="4">f(7,16)</td></tr>
<tr><td colspan="4">mid = (1 *2 + 6) / 3 = 2</td><td colspan="4">mid = (7*2+16)/3 = 10</td></tr>
<tr><td colspan="4">f(1,6)=f(1,2)+f(3,6)+1</td><td colspan="4">f(7,16) = f(7,10) + f(8,16)+1</td></tr>
<tr><td colspan="2">f(1,2)</td><td colspan="2">f(3,6)</td><td colspan="2">f(7,10)</td><td colspan="2">f(8,16)</td></tr>
<tr><td colspan="2">mid = 1</td><td colspan="2">mid = 4</td><td colspan="2">mid = 8</td><td colspan="2">mid = 10</td></tr>
<tr><td colspan="2">f(1,1)+f(2,2)+1</td><td colspan="2">f(3,4)+f(5,6)+1</td><td colspan="2">f(7,8)+f(9,10)+1</td><td colspan="2">f(8,10)+f(11,16)+1</td></tr>
<tr><td>return 1</td><td>return 1</td><td>f(3,4)</td><td>f(5,6)</td><td>f(7,8)</td><td>f(9,10)</td><td>f(8,10)</td><td>f(11,16)</td></tr>
<tr><td></td><td></td><td>mid 3</td><td>mid 4</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

1부터 16까지의 수의 mid를 구하여 a, b가 같은 수가 될 때까지 나누어 재귀함수를 호출하는 코드이다. 1부터 16까지는 16개의 수가 있고 1부터 16까지 나누는 부분은 15개가 존재한다. 따라서 답은 16+15가 정답이다.

34. 다음 프로그램의 출력 결과는 무엇인가?

```
int A[5]={1,3,4,7,8};
int B[5]={2,5,6,8,9};
int i=0, j=0;
while(i<5 || j<5){
  if (j >= 5 || (i<5 && A[i]<B[j])) printf("%d", A[i++]);
  else printf("%d", B[j++]);
}
printf("\n");
```

① 123456789 ② 1347825689 ③ 2568913478
④ 1235467889 ⑤ 1234567889

** 출제의도?
코드의 이해

** 의미?
  배열 참조 계산
** 해결예시?

순서	j >= 5	(i < 5 && A[i] < B[j]	i	j	printf("%d",)
시작			0	0	
1	거짓	참	1	0	1
2	거짓	거짓	1	1	2
3	거짓	참	2	1	3
4	거짓	참	3	1	4
5	거짓	거짓	3	2	5
6	거짓	거짓	3	3	6
7	거짓	참	4	3	7
8	거짓	거짓	4	4	8
9	거짓	참	5	4	8
10	거짓	거빗	5	5	9

35. 다음 프로그램의 출력 결과는 무엇인가?

```
int g(int x){
  if(x<10) return x;
  else return g(x/10)+(x%10);
}
int main(){
  int i s=0;
  for(i=1; i<=100; i++) s+=g(i);
  printf("%d\n", s);
  return 0;
}
```

① 899 ② 900 ③ 901 ④ 902 ⑤ 903

** 출제의도?
  코드의 이해
** 의미?
  재귀 함수의 이해,
** 해결예시?
  재귀함수 g(x)는 어떤 정수 x의 자릿수를 분리하여 합을 구하는 함수이다. 이 함수를 반복문을 통하여 1~100까지 반복하고 있으므로, 결국 구하는 변수 s=g(1)+g(2)+...+g(100)을 계산한 결과를 묻는 것이다. 이것을 1부터 100까지 일일이 자릿수를 분리하여 더한다면 시간이 너무 많이 걸릴 것이므로, 단순화해서 계산하는 능력이 필요하다.
  1) 일의 자리 숫자 0~9가 10번 반복되므로, 0~9의 합 * 10 = 45*10 = 450이다.
  2) 십의 자리 숫자 1~9가 10번씩 나오므로, (1*10)+(2*10)+...+(9*10)인데, 이것은 결국 1)번 경우와 같은 경우이므로 450이다.
  3) 백의 자리 숫자는 딱 100이 되었을 때 한 가지 경우이므로, 1이다.

따라서 1)+2)+3) = 450+450+1 = 901이다.

다른 풀이

함수를 살펴보자. 만약 int g(99)를 호출하면

g(99) = g(99/10) + 99%10	->	g(99/10)	->	rerutn 9
	->	99%10=9		

따라서 g(99) 함수는 18=9+9를 반환한다. 예와 같이 이 함수는 각 자리 수의 합을 반환하는 함수이다.

출력하는 값은 1부터 100까지 총 1의 개수, 2의 개수, 3의 개수를 모두 구하여 더하면 된다.

숫자		개수		합
1	X1(X는 0부터 9까지)과 1X(X는 0부터 9까지) +1(숫자 100)	21	21 * 1	21
2	X2(X는 0부터 9까지)과 2X(X는 0부터 9까지)	20	20 * 2	40
3	X3(X는 0부터 9까지)과 3X(X는 0부터 9까지)	20	20 * 3	60
4	X4(X는 0부터 9까지)과 4X(X는 0부터 9까지)	20	20 * 4	80
5	X5(X는 0부터 9까지)과 5X(X는 0부터 9까지)	20	20 * 5	100
6	X6(X는 0부터 9까지)과 6X(X는 0부터 9까지)	20	20 * 6	120
7	X7(X는 0부터 9까지)과 7X(X는 0부터 9까지)	20	20 * 7	140
8	X8(X는 0부터 9까지)과 8X(X는 0부터 9까지)	20	20 * 8	160
9	X9(X는 0부터 9까지)과 9X(X는 0부터 9까지)	20	20 * 9	180
총합				901

36. 다음 프로그램의 출력 결과는 무엇인가?

```
int a=2, b=4;
while(a+b<110){
  int c=b+a;
  a=b;
  b=c;
}
printf("%d %d\n", a, b);
```

① 42 68　　② 68 110　　③ 110 178
④ 32 64　　⑤ 64 128

** 출제의도?
코드의 이해
** 의미?
트레이싱

** 해결예시?

코드가 어렵지 않지만 변수 a, b, c의 값의 변화를 잘 관찰하려면 표를 이용하는 것이 좋다.

c		6	10	16	26	42	68
a	2	4	6	10	16	26	42
b	4	6	10	16	26	42	68

따라서 a와 b의 값은 42, 68이다.

다른 풀이

과정을 아래의 표에서와 같이 진행된다.

과정	c = a + b	a	b
초기 값		2	4
1	6	4	6
2	10	6	10
3	16	10	16
4	26	16	26
5	42	26	42
6	68	42	68
7	110		

a(i)=a(i-2)+a(i-1) (i>=2) 의 값과 같이 피보나치수열과 비슷하게 진행된다.

37. 다음 프로그램은 두 자연수 a, b의 최대공약수를 구하는 함수이다. 빈 칸에 들어갈 코드는 무엇인가?

```
int gcd(int a, int b){
  if(b==0) return a;
  return [          ];
}
```

① gcd(b % a, a)　② gcd(a / b, b)　③ gcd(a % b, b)
④ gcd(b, a / b)　⑤ gcd(b, a % b)

** 출제의도?

코드의 이해

** 의미?

유클리드 호제법의 구현, 재귀 방법의 이해

** 해결예시?

두 자연수 a, b의 최대공약수를 구하는 gcd(a, b) 함수에서 b가 0일 때 a를 반환하는 것을 보면, 유클리드 호제법 알고리즘이다.

a가 b로 나누었을 때 나머지가 생긴다면, b를 그 나머지로 나눈다. 이와 같은 방법을 계속 반복하여 나머지가 0이 되었을 때, 마지막에 나누던 수가 두 수의 최대공약수가 된다. 따라서 이를 재귀적인 함수로 표현하면 다음과 같다.

```
int gcd(int a, int b) {
  if(b==0) return 0;
```

```
  return  gcd( b, a%b )  ;
}
```

38. 다음 프로그램의 출력 결과는 무엇인가?

```
int g(int n){
  return (n*n+1)%9;
}
int f(int x, int y){
  int a=g(x), b=g(g(y));
  if(a==b) return 1;
  return 1+f(a, b);
}
int main(){
  int s=0, i;
  for(i=0; i<9; i++) s+=f(i, i);
  printf("%d\n", s);
  return 0;
}
```

① 9　　② 18　　③ 21　　④ 24　　⑤ 27

** 출제의도?
  코드의 이해
** 의미?
  재귀 함수의 이해
** 해결예시?
  변수 s에 f(0, 0)부터 f(8, 8)까지의 값을 누적하여 화면에 출력하여야 한다.
  f(x, y) 함수는 g(x)와 g(g(y)) 함수의 값이 서로 같을 때는 1을 반환하고, 그렇지 않은 경우에는 g(x)+g(g(y))+1을 반환하는 함수이다. f(x, y) 함수의 매개변수로 올 수 있는 값은 0부터 8까지이므로 계산을 편리하게 하기 위하여 g(0)부터 g(8)까지, 그리고 g(g(0))부터 g(g(8))까지의 값을 미리 정리하여 보자. 또한 한 번 계산한 값은 나중에 다시 계산하지 않고 바로 이용하여 계산 시간을 줄이도록 한다.

g(0)=(0^2+1)%9=1	g(g(0))=g(1)=2	f(0, 0)=f(1, 2)+1 = f(2, 8)+1+1 = 1+1+1 = 3
g(1)=(1^2+1)%9=2	g(g(1))=g(2)=5	f(1, 1)=f(2, 5)+1 = f(5, 2)+1+1 = 1+1+1 = 3
g(2)=(2^2+1)%9=5	g(g(2))=g(5)=8	f(2, 2)=f(5, 8)+1 = f(8, 5)+1+1 = 1+1+1 = 3
g(3)=(3^2+1)%9=1	g(g(3))=g(1)=2	f(3, 3)=f(1, 2)+1 = 3
g(4)=(4^2+1)%9=8	g(g(4))=g(8)=2	f(4, 4)=f(8, 2)+1 = f(2, 8)+1+1 = 1+1+1 = 3
g(5)=(5^2+1)%9=8	g(g(5))=g(8)=2	f(5, 5)=f(8, 2)+1 = 3
g(6)=(6^2+1)%9=1	g(g(6))=g(1)=2	f(6, 6)=f(1, 2)+1 = 3
g(7)=(7^2+1)%9=5	g(g(7))=g(5)=8	f(7, 7)=f(5, 8)+1 = 3
g(8)=(8^2+1)%9=2	g(g(8))=g(2)=5	f(8, 8)=f(2, 5)+1 = 3

  위와 같이 정리하여 보면 f(0, 0)부터 f(8, 8)까지의 반환 값이 모두 3인 것을 알 수 있

다. 따라서 변수 s에 저장되는 값은 f(1, 1) + f(2, 2) + ... + f(8, 8) = 3 * 9 = 27이다.

39. 다음 프로그램의 출력 결과는 무엇인가?

```
int n=2015, i=1, tot=0;
int a=0;
while(n>0){
  if(n%10==2) tot+=(n/10)*i+(a+1);
  else if(n%10>2) tot+=(n/10+1)*i;
  else tot+=(n/10)*i;
  a+=i*(n%10);
  i*=10;
  n/=10;
}
printf("%d\n", tot);
```

① 616　② 617　③ 618　④ 619　⑤ 620

** 출제의도?
코드의 이해
** 의미?
트레이싱
** 해결예시?
이 프로그램의 로직을 이해하는 것이 중요한데, n의 작은 자릿수부터 하나씩 분리하여 해당 조건에 맞는 연산을 tot에 누적하고 있다. 변수 a는 분리된 값을 저장하고, i는 자릿값을 나타낸다. 제어문을 따라 변수의 값을 정리해 보면 다음과 같다.

n	a	i	tot
2015	0	1	202
201	5	10	402
20	15	100	602
2	15	1000	618

따라서 tot는 618이다.

40. 다음 프로그램의 출력 결과는 무엇인가?

```
int n=2015, c=0;
while(n>0){
  c+=(n&3);
  n=n&(n-1);
}
printf("%d\n", c);
```

① 0　　② 2　　③ 4　　④ 5　　⑤ 8

** 출제의도?
코드의 이해
** 의미?
비트연산자 &의 이해
** 해결예시?
n=2015를 이진수로 나타내면 111 1101 1111이다.
1) n&3의 의미: 3은 이진수로 11이다. 어떤 수n과 3을 AND연산하면 n의 최하위 2비트만 남는 효과가 있다.

```
111 1101 1111
000 0000 0011
-------------
000 0000 0011
```

2) n&(n-1)의 의미 : 어떤 수 n과 1작은 수 n-1을 AND연산하면, n의 1이 있는 최하위 비트가 0으로 되는 효과가 있다.

```
111 1101 1111
111 1101 1110
-------------
111 1101 1110
```

```
111 1101 1110
111 1101 1101
-------------
111 1101 1100
```

```
111 1101 1100
111 1101 1011
-------------
111 1101 1000
```

```
111 1101 1000
111 1101 0111
-------------
111 1101 0000
```

n은 반복문에서 2)에 의해 2015 -> 2014 -> 2012-> 2008-> 2000-> 1984 -> ... -> 1024 -> 0이 된다. n의 값을 일일이 따라 가려면 시간이 많이 걸린다. 하지만 1)번 연산을 잘 생각해보면 최하위 2비트만 계속 AND연산을 하고 있으므로, n=2015일 때 c=3이고, n=2014일 때, c=5가 된다. 그리고 그 이후 n에 대해서는 1)번 연산의 결과가 무조건 0이므로 c=5가 유지되므로 계산할 필요가 없다. 따라서 정답은 c=5이다.

41. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, tot=0;
int a[4]={0,2,3,1}, b[4]={9,1,3,9};
int c[4]={1,3,6,0};
for(i=0; i<8; i++){
  tot+=a[(i+1)%4]*b[(i+3)%4];
  tot-=b[(i*3)%4]*c[(i+2)%4];
  tot+=a[i%4]*c[i%4];
}
printf("%d\n", tot);
```

① 20　　② -20　　③ 12　　④ 24　　⑤ -24

** 출제의도?
코드의 이해
** 의미?

비트연산자 &의 이해
** 해결예시?

순	i	a[(i+1)%4]	b[(i+3)%4]	b[(i*3)%4]	c[(i+2)%4]	a[i%4]	c[i%4]	tot
시작								0
1	0	2	9	9	6	0	1	-36
2	1	3	9	9	0	2	3	-3
3	2	1	1	3	1	3	6	13
4	3	0	3	1	3	1	0	10
5	4	2	9	9	6	0	1	-26
6	5	3	9	3	0	2	3	7
7	6	1	1	3	1	3	6	23
8	7	0	3	1	3	1	0	20

i = 0 일 때 2 * 9 - 9 * 6 + 0 * 1 = -36
i = 1 일 때 3 * 9 - 3 * 0 + 2 * 3 = 33
i = 2 일 때 1 * 1 - 3 * 1 + 3 * 6 = 16
i = 3 일 때 0 * 3 - 1 * 3 + 1 * 0 = -3
i = 4 일 때 2 * 9 - 9 * 9 + 6 * 0 = -36
i = 5 일 때 3 * 9 - 3 * 0 + 2 * 3 = 33
i = 6 일 때 1 * 1 - 3 * 1 + 3 * 6 = 16
i = 7 일 때 0 * 3 - 1 * 3 + 1 * 0 = -3

42. 다음 프로그램의 두 번째 출력 결과는 001011 이다. 일곱 번째 출력 결과는 무엇인가?

```
#include <stdio.h>
int F(int *A, int n){
  int i, prev=0, cnt1=0;
  for(i=n-1; i>=0; i--){
    if(A[i]<prev){
      A[i++]=1;
      A[i++]=0;
      for(; i<n; i++) A[i]=(--cnt1>0);
      return 1;
    }
    if(A[i]==1) cnt1++;
    prev=A[i];
  }
  return 0;
}
int main(){
  int A[6]={0,0,0,1,1,1};
```

```
  do{
    int i;
    for(i=0; i<6; i++) printf("%d", A[i]);
    printf("\n");
  }while(F(A, 6));
  return 0;
}
```

① 000000 ② 111111 ③ 010110
④ 011001 ⑤ 011100

** 출제의도?
코드의 이해
** 의미?
수의 이동
** 해결예시?
함수 F의 동작을 이해하는 문제이다. 함수 F는 배열 A의 1을 좌측으로 한 칸씩 이동하는 함수인데 다음과 같이 동작한다.
000111 → 001011 → 001101 → 001110 → 010110 →
011010 → 011100 → 101100 → 110100 → 111000
따라서 일곱 번째 출력 결과는 011100이다.

43. 다음 프로그램의 출력 결과로 출력되지 않는 숫자는 무엇인가?

```
int a[3][3]={1,2,3,1,3,2,0,1,2}, b[3][3]={1,3,2,0,1,1,2,3,1};
int main(){
  int k, i, j;
  for(k=0; i<3; k++){
    for(i=0; i<3; i++){
    int s=0;
    for(j=0; j<3; j++) s+=a[k][j]*b[j][i];
      printf("%d\n", s);
    }
  }
}
```

① 3 ② 2 ③ 5 ④ 7 ⑤ 14

** 출제의도?
코드의 이해
** 의미?
행렬의 곱
** 해결예시?

a[k][j]	j=0	1	2
k=0	1	2	3
1	1	3	2
2	0	1	2

b[j][i]	i=0	1	2
j=0	1	3	2
1	0	1	1
2	2	3	1

k=0, i=0이면 1*1+2*0+3*2=7이다. 계산 결과를 보면 행렬의 곱과 관련 있다는 것을 볼 수 있다. 따라서 행렬의 곱은 $\begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \\ 0 & 1 & 2 \end{pmatrix}\begin{pmatrix} 1 & 3 & 2 \\ 0 & 1 & 1 \\ 2 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 7 & 14 & 7 \\ 5 & 12 & 7 \\ 4 & 7 & 3 \end{pmatrix}$ 이다.

따라서 나올 수 없는 값은 2이다.

44. 다음 프로그램의 출력 결과는 무엇인가?

```
void f(int t, int mat[2][2]){
  int i, j;
  int tmp[2][2];
  for(i=0; i<2; i++){
    for(j=0; j<2; j++){
      if(t%2==0) tmp[i][j]=mat[1-i][j];
      else tmp[i][j]=mat[i][1-j];
    }
  }
  for(i=0; i<2; i++){
    for(j=0; j<2; j++) mat[i][j]=tmp[i][j];
  }
}
int main(){
  int mat[2][2]={{2,0}, {1,5}};
  int i, j;
  for(i=0; i<2; i++) f(i, mat);
  for(i=0; i<2; i++){
    for(j=0; j<2; j++) printf("%d ", mat[i][j]);
    printf("\n");
  }
  return 0;
}
```

① $\begin{matrix}2 & 0\\ 1 & 5\end{matrix}$ ② $\begin{matrix}0 & 2\\ 5 & 1\end{matrix}$ ③ $\begin{matrix}1 & 5\\ 2 & 0\end{matrix}$ ④ $\begin{matrix}5 & 1\\ 2 & 0\end{matrix}$ ⑤ $\begin{matrix}5 & 1\\ 0 & 2\end{matrix}$

** 출제의도?
 코드의 이해
** 의미?
 패턴 찾기
** 해결예시?
 함수 f(t, MAT)의 동작을 정의해보면, t가 짝수일 때 배열 원소의 상하를 교환하고, t가 홀수일 때 배열 원소의 좌우를 교환한다.

2 0 1 5	1 5 2 0	5 1 0 2	0 2 5 1
MAT 배열	t=0	t=1	t=2
2 0 1 5	1 5 2 0	5 1 0 2	0 2 5 1
t=3	t=4	t=5	t=6

4번의 동작으로 배열이 처음 상태대로 돌아온다. i=2014일 때 상태를 계산해보면
2014 % 4 = 2
 가 되므로 t=2일 때의 상태와 동일하다.

45. 다음 프로그램의 출력 결과는 무엇인가?

```
int a[10]={5,4,2,9,1,3,8,10,6,7};
int main(){
  int i, j, h = 9, n = 10;
  while(h>=1){
    for(i=h; i<n; i++){
      int temp=a[i];
      for(j=i-h; j>=0; j-=h){
        if(a[j]>temp) a[j+h]=a[j];
        else break;
      }
      a[j+h] = temp;
    }
    h/=3;
  }
  for(i=0; i<n; i++) printf("%d ", a[i]);
}
```

① 1 2 3 4 5 6 7 8 9 10 ② 1 2 3 4 5 6 8 9 10 7

③ 1 5 8 2 6 9 3 7 10 4　　　　　　④ 5 4 3 2 1 10 9 8 7 6

⑤ 10 9 8 7 6 5 4 3 2 1

** 출제의도?
코드의 이해
** 의미?
쉘 정렬
** 해결예시?
쉘 정렬코드이다. 오름차순 정렬을 하고 있다. 따라서 답은 1 2 3 4 5 6 7 8 9 10이다.

46. 다음 프로그램의 출력 결과는 무엇인가?

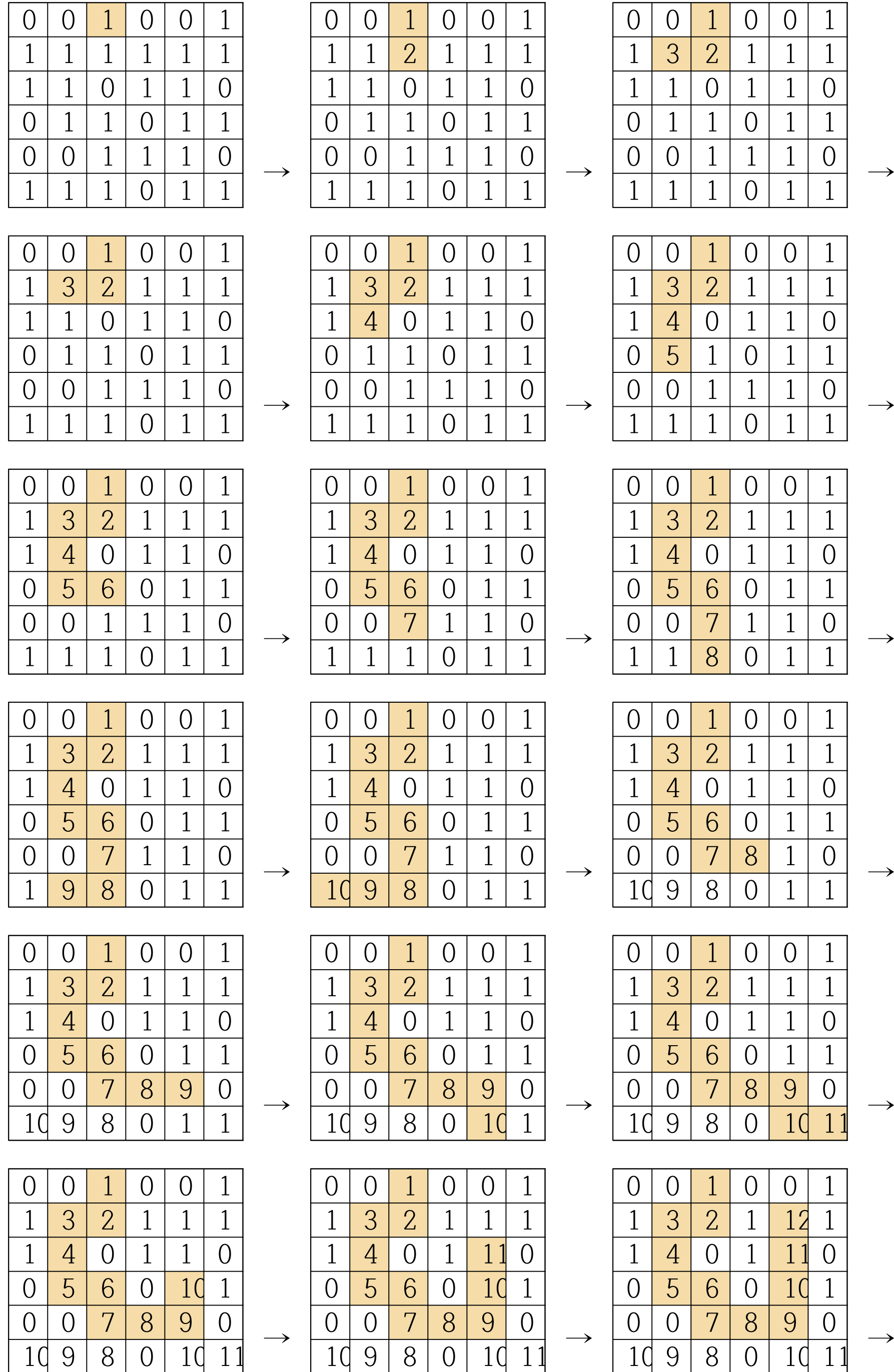
```
char a[7][7]={"001001","111111","110110","011011","001110","111011"};
int tot=0;
void f(int n, int m, int c){
  if(n<0 || m<0 || n>=6 || m>=6) return;
  if(a[n][m]=='0') return;
  tot+=c;
  a[n][m]='0';
  f(n-1, m, c+1);
  f(n+1, m, c+1);
  f(n, m-1, c+1);
  f(n, m+1, c+1);
}
int main(){
  int i, j;
  for(i=0; i<6; i++){
    for(j=0; j<6; j++){
      if(a[i][j]=='1') f(i, j, 1);
    }
  }
  printf("%d\n", tot);
  return 0;
}
```

① 24　　② 144　　③ 180　　④ 196　　⑤ 202

** 출제의도?
코드의 이해
** 의미?
깊이우선탐색(Depth First Search)
** 해결예시?

함수 f(n, m, c)의 DFS 순서를 묻는 문제이다.

함수 f(n, m, c)는 '1'을 깊이 우선 탐색(DFS)한다. 탐색 순서는 코드를 분석해보면, 현재 위치에서 상, 하, 좌, 우 순서로 탐색하는 것을 알 수 있다. 시작위치는 (1, 3)이다.



0	0	1	0	0	1
1	3	2	13	12	1
1	4	0	1	11	0
0	5	6	0	10	1
0	0	7	8	9	0
10	9	8	0	10	11

→

0	0	1	0	0	1
1	3	2	13	12	1
1	4	0	14	11	0
0	5	6	0	10	1
0	0	7	8	9	0
10	9	8	0	10	11

→

0	0	1	0	0	1
1	3	2	13	12	13
1	4	0	14	11	0
0	5	6	0	10	1
0	0	7	8	9	0
10	9	8	0	10	11

→

0	0	1	0	0	14
1	3	2	13	12	13
1	4	0	14	11	0
0	5	6	0	10	1
0	0	7	8	9	0
10	9	8	0	10	11

→

0	0	1	0	0	14
1	3	2	13	12	13
1	4	0	14	11	0
0	5	6	0	10	11
0	0	7	8	9	0
10	9	8	0	10	11

→

0	0	1	0	0	14
1	3	2	13	12	13
5	4	0	14	11	0
0	5	6	0	10	11
0	0	7	8	9	0
10	9	8	0	10	11

→

0	0	1	0	0	14
6	3	2	13	12	13
5	4	0	14	11	0
0	5	6	0	10	11
0	0	7	8	9	0
10	9	8	0	10	11

<최종 상태>

tot는 호출시 c의 값을 누적한 것이므로, 최종 상태에 나타난 모든 숫자를 더한 값과 같다. 따라서 tot=202이다.

47. 다음 프로그램의 출력 결과는 무엇인가?

```
int i, j, tot=0, a[21]={1,};
for(i=2; i<=20; i++){
  if(a[i]) continue;
  for(j=i; j<=20; j+=i) a[j]+=j/i;
}
for(i=0; i<=20; i++) tot+=a[i];
printf("%d\n", tot);
```

① 26　　② 27　　③ 93　　④ 94　　⑤ 114

** 출제의도?

코드의 이해

** 의미?

에라토스테네스의 체

** 해결예시?

이 코드는 에라토스테네스의 체(Sieve of Eratosthenes) 알고리즘을 변형한 것이다. 에라토스테네스의 체는 특정 구간의 소수를 찾는 방법 중 효율이 매우 좋은 알고리즘이다. 어떤 수가 소수이면 이 수의 배수는 소수가 될 수 없다는 것에 아이디어를 얻어서, 소수의 배수는 모두 소수가 아님을 체크하는 기법이다.

약간 변형된 이 코드로 알고리즘을 표현해보자.

1) i=2일 때

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
a	1		1		2		3		4		5		6		7		8		9		10

2) i=3일 때

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
a	1		1	1	2		3 + 2		4	3	5		6 + 4		7	5	8		9 + 6		10

3) i=5일 때

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
a	1		1	1	2	1	5		4	3	5 + 2		10		7	5 + 3	8		15		10 + 4

4) i=7일 때

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
a	1		1	1	2	1	5	1	4	3	7		10		7 + 2	8	8		15		14

5) i=11일 때

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
a	1		1	1	2	1	5	1	4	3	7	1	10		9	8	8		15		14

6) i=13일 때

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
a	1		1	1	2	1	5	1	4	3	7	1	10	1	9	8	8		15		14

7) i=17일 때

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
a	1		1	1	2	1	5	1	4	3	7	1	10	1	9	8	8	1	15		14

8) i=11일 때

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
a	1		1	1	2	1	5	1	4	3	7	1	10	1	9	8	8	1	15	1	14

이렇게 배열이 채워지며, 문제에서 요구하는 tot는 이 배열을 모두 더한 값인 94가 된다.

48. ccw(0, 0, 1, 0, 1, 1)의 결과는 1이다. 다음 프로그램의 출력 결과는 무엇인가?

```
#include <stdio.h>
int ccw(int x1, int y1, int x2, int y2, int x3, int y3){
  return (x2-x1)*(y3-y1)-(x3-x1)*(y2-y1);
}
int main(){
  int A[5] = {0,3,1,2,2};
  int B[5];
```

```
  int i, cnt=0;
  for(i=0; i<5; i++){

   while(cnt>=2&&ccw(B[cnt-2],A[B[cnt-2]],B[cnt-1],A[B[cnt-1]],i,A[i])>=0)
    cnt--;
    B[cnt++]=i;
  }
  for(i=0; i<cnt; i++) printf("%d", B[i]);
  printf("\n");
  return 0;
}
```

① 01234 ② 01 ③ 0134 ④ 014 ⑤ 02341

** 출제의도?
코드의 이해
** 의미?
계산 기하
** 해결예시?
ccw함수는 평면위의 세 점이 반시계 방향인지 판별하는 함수이다. 반시계 방향이면 음의 정수, 시계 방향이면 양의 정수를 리턴한다. 반복 조건은 cnt가 2이상이고 ccw가 시계 방향이면, cnt를 1감소시키고, ccw가 반시계방향이 될 때까지 반복한다. 이 코드로 진행되는 값의 상태는 다음과 같다.

i	ccw	배열 B
0		0
1		0 1
2	-5	0 1 2
3	3 -> -7	0 1 3
4	1 -> -10	0 1 4

따라서 정답은 014이다.

[49-50] 다음과 같은 문제를 해결하기 위해 프로그램을 작성하였다. 물음에 답하시오.

무게가 서로 다른 N개의 물건이 있다. 각 물건은 1부터 N까지 번호가 매겨져 있다. 우리는 일부 물건 쌍에 대해서 양팔 저울로 어떤 것이 무거운 것인지를 측정한 결과표를 가지고 있다. 이 결과표로부터 직접 측정하지 않은 물건 쌍의 비교 결과를 알아낼 수도 있고 알아내지 못할 수도 있다.

예를 들어, 총 6개의 물건이 있고, 다음 5개의 비교 결과가 주어졌다고 가정하자.([1]은 1번 물건의 무게를 의미한다.)

[1] > [2], [2] > [3], [3] > [4], [5] > [4], [6] > [5]

우리는 [2] > [3], [3] > [4]로부터 [2] > [4]라는 것을 알 수 있다. 하지만, 물건 2와 물건 6을 비교하는 경우, 앞서의 결과만으로는 어느 것이 무거운지 알 수 없다. 이와 같이, 물건 2는 물건 1, 3, 4와의 비교 결과는 알 수 있지만, 물건 5, 6과의 비교 결과는 알 수 없다. 물건 4는 모든 다른 물건과의 비교 결과를 알 수 있다.

비교 결과가 모순되는 입력은 없다고 가정한다. 위 예제의 기존 측정 결과에 [3] > [1]이 추가되었다고 가정하자. 이 경우 [1] > [2], [2] > [3]이므로 우리는 [1] > [3]이라는 것을 예측할 수 있는데, 이는 기존에 측정된 결과 [3] > [1]과 서로 모순이므로 이러한 입력은 가능하지 않다.

물건의 개수 N과 일부 물건 쌍의 비교 결과가 주어졌을 때, 각 물건에 대해서 그 물건과의 비교 결과를 알 수 없는 물건의 개수를 출력하는 프로그램을 작성하시오.

입력 형식

표준입력의 첫 줄에는 물건의 개수 N이 주어지고, 둘째 줄에는 미리 측정된 물건 쌍의 개수 M이 주어진다. 단, $5 \le N \le 100$ 이고, $0 \le M \le 2{,}000$이다. 다음 M개의 줄에 미리 측정된 비교 결과가 한 줄에 하나씩 주어진다. 각 줄에는 측정된 물건 번호를 나타내는 두 개의 정수가 공백을 사이에 두고 주어지며, 앞의 물건이 뒤의 물건보다 더 무겁다.

출력 형식

여러분은 N개의 줄에 결과를 출력해야 한다. i 번째 줄에는 물건 i와 비교 결과를 알 수 없는 물건의 개수를 출력한다.

입력과 출력의 예 1

입력

```
6
5
1 2
2 3
3 4
5 4
6 5
```

출력

```
2
2
2
0
3
3
```

입력과 출력의 예 2

입력

```
9
11
2 1
3 1
2 8
2 9
7 8
4 5
6 7
6 3
1 7
6 2
1 9
```

출력

```
2
3
3
7
7
2
3
3
4
```

프로그램

```
#include <stdio.h>
int f[101][101];
int main(){
  int N, M;
  scanf("%d %d", &N, &M);
  for(int i=0, x, y; i<M; i++){
    scanf("%d %d", &x, &y);
    f[x][y]=1;
  }
  for(int k=1; i<=N; k++){
    for(int i=1; i<=N; i++){
      for(int j=1; j<=N; j++){
        if(   (a)   ) f[i][j]=1;
      }
    }
  }
  for(int i=1; i<=N; i++){
    int count=0;
    for(int j=1; j<=N; j++)
      if(i!=j &&    (b)   ) count++;
    printf("%d\n", count);
  }
  return 0;
}
```

49. 다음 중 (a)에 들어갈 내용으로 알맞은 것은?

① f[i][k] ^ f[k][j]　② f[i][k] || f[k][j]　③ f[k][j] || f[j][i]
④ f[i][k] && f[k][j]　⑤ f[k][j] && f[j][i]

50. 다음 중 (b)에 들어갈 내용으로 알맞은 것은?

① (f[i][j] | f[j][i]) == 0　② f[i][j] * f[j][i] == 0
③ (f[i][j] ^ f[j][i]) == 1　④ f[i][j] && f[j][i]
⑤ f[i][j] == 0

** 출제의도?

코드의 이해
** 의미?
문제 해결, 반사/대칭/추이 관계, 알고리즘
** 해결예시?
문제의 상태를 2차원 배열 자료구조 f로 나타내어 보자. 여기서 1은 [x]번 물건과 [y]번 물건의 관계를 알 수 있는 상태를 의미한다.
1) 먼저 입력의 상태를 배열로 나타내어 보면,

f	[1]	[2]	[3]	[4]	[5]	[6]
[1]		1				
[2]			1			
[3]				1		
[4]						
[5]				1		
[6]					1	

이다.
2) 이 상태에서 추이적으로 알 수 있는 상태를 체크(3중 for 부분)해보면,

f	[1]	[2]	[3]	[4]	[5]	[6]
[1]		1	1	1		
[2]			1	1		
[3]				1		
[4]						
[5]				1		
[6]				1	1	

이다. 알고리즘 구현 시 설계를 어떻게 하느냐에 따라 인덱스를 다양한 방법으로 구현할 수 있지만, 문제에서 f[i][j]=1의 힌트가 있으므로 if문의 조건으로는 (f[i][k] && f[k][j])가 되어야 한다.
3) 반사관계와 대칭관계를 체크해서 알 수 없는 물건의 개수를 세고 출력한다.
a) 반사관계는 자기 자신을 가리키는 경우이므로 당연히 알 수 있는 관계이고, 반사관계를 제외하려면 조건문에서 (i != j)로 쉽게 표현할 수 있다.
b) 대칭관계는 f[x][y]의 관계를 알면 f[y][x]의 관계도 바로 알 수 있는 관계이다. 대칭관계를 제외하려면 조건문에서 (f[i][j] | f[j][i])==0 으로 나타낼 수 있다. 관계 OR연산을 사용해도 되지만 비트 OR연산을 사용하였다. 결과는 같다.
물건의 무게를 알 수 없는 경우는 a), b)조건을 모두 만족하는(AND) 경우이다. 이 물건들의 개수를 세고 출력한다.
참고로 반사/대칭/추이 관계에 있는 물건들을 모두 체크한 배열의 상태이다.

f	[1]	[2]	[3]	[4]	[5]	[6]	개수
[1]	1	1	1	1			**2**
[2]	1	1	1	1			**2**
[3]	1	1	1	1			**2**
[4]	1	1	1	1	1	1	**0**
[5]				1	1	1	**3**
[6]				1	1	1	**3**